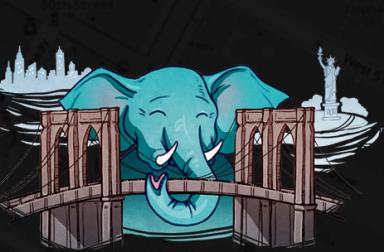


PGCONF NYC 2023



# MAP OF AMAZING POSTGRES EXTENSIONS YOU MIGHT NOT KNOW ABOUT

CLAIRE GIORDANO

@clairegiordano@hachyderm.io • @clairegiordano • @AzureDBPostgres • @citusdata

Engineer  
Dev Manager

PM

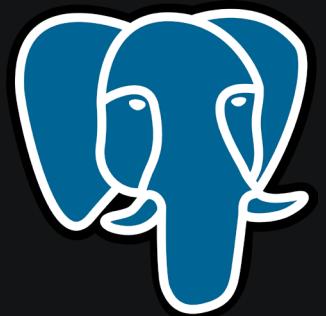
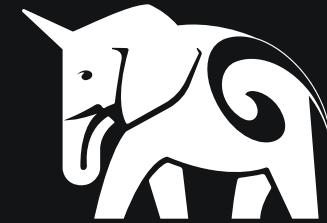
Marketer  
Writer

Open-source champion  
Community Lead



@clairegiordano

@clairegiordano@hachyderm.io



# Citus = Postgres open-source extension that first got me involved with PG

The screenshot shows the GitHub repository page for `citusdata / citus`. The repository is public and has 598 branches and 144 tags. The `Code` tab is selected. A recent commit by `onderkalaci` is highlighted, adding a missing image to the Technical Readme. The commit was made yesterday at `0dca65c` and has 6,718 commits. The repository has 897 issues, 65 pull requests, 212 watchers, and 596 forks. A callout highlights the `Starred` button with 9k stars. To the right, a vertical sidebar lists several yellow star icons. The repository description is "Distributed PostgreSQL as an extension". It includes links to [www.citusdata.com](http://www.citusdata.com), [Readme](#), [AGPL-3.0 license](#), and [Code of conduct](#). The repository uses the `postgres`, `multi-tenant`, `sql`, `database`, `scale`, `postgresql`, `sharding`, `distributed-database`, `citus`, `relational-database`, `database-cluster`, and `citus-extension` tags.

citusdata / citus

Type  to search

Code Issues 897 Pull requests 65 Discussions Actions Wiki Security Insights

citus Public Edit Pins Watch 212 Fork 596 Starred 9k

main 598 branches 144 tags Go to file Add file Code

onderkalaci Addd missing image to ... 0dca65c yesterday 6,718 commits

.circleci Bump cryptography from 41.0.3 to 41.0.4 in ... 4 days ago

.github Removes centos 7 for PG 16 in packaging pi... 2 weeks ago

ci Fix string eval bug in migration files check (...) 7 months ago

config Renames configure.in to fix warnings (#6034) last year

images Addd missing image to Technical Readme (...) yesterday

src Addd missing image to Technical Readme (...) yesterday

vendor Update cherry-pick hash in vendor README 3 years ago

About

Distributed PostgreSQL as an extension

[www.citusdata.com](http://www.citusdata.com)

postgres multi-tenant sql database

scale postgresql sharding

distributed-database citus

relational-database database-cluster

citus-extension

Readme

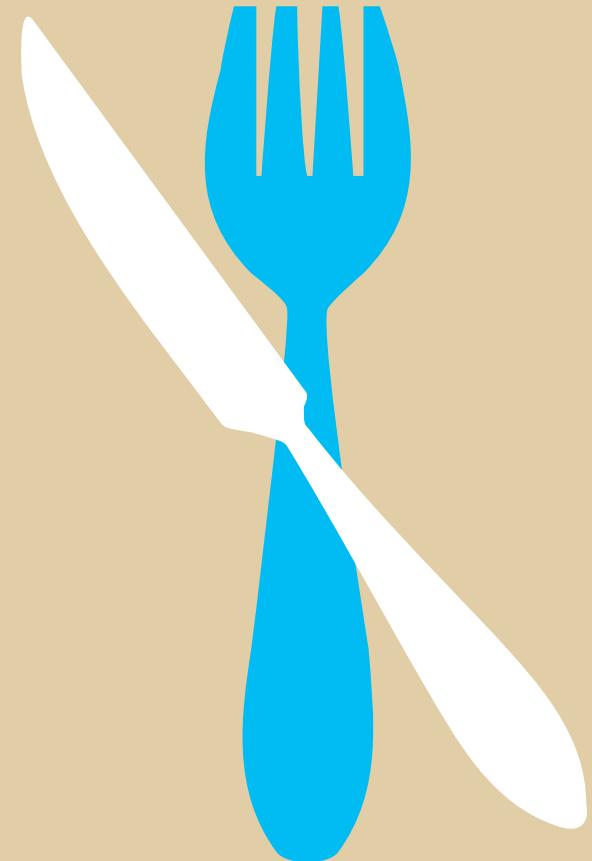
AGPL-3.0 license

Code of conduct

# What is a Postgres extension?

- Adds functionality to Postgres  
(hence: “extends” Postgres)
  - Not a fork
  - To use—install Postgres & the PG extension(s)
  - To enable (or disable), get familiar with

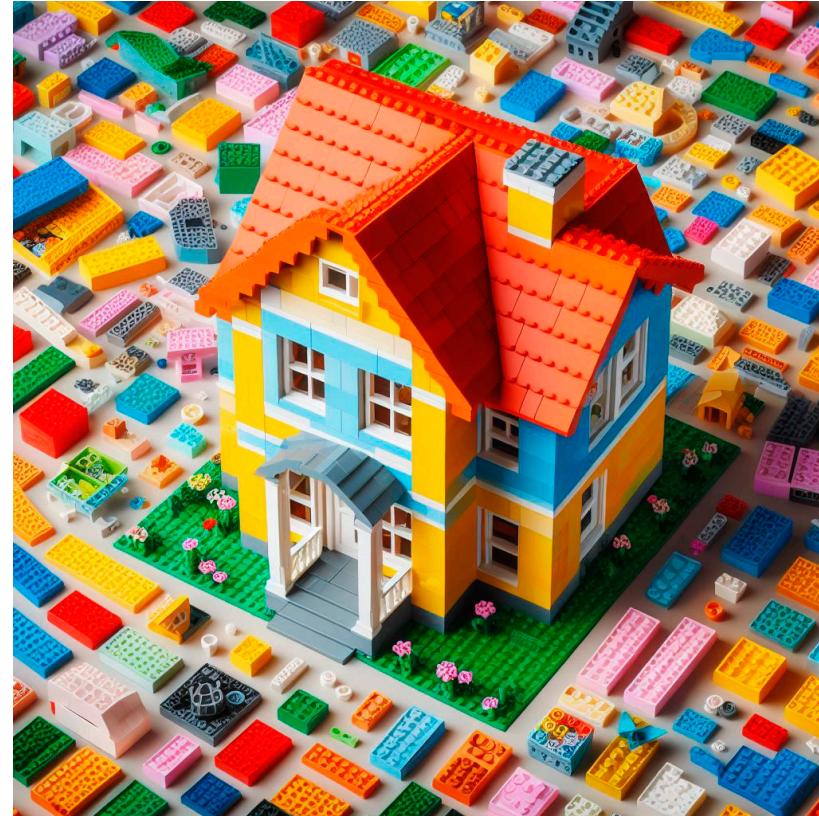
**CREATE EXTENSION**  
**ALTER EXTENSION**  
**DROP EXTENSION**



# Search for a good analogy for Postgres Extensions



plug & play



beloved modular building toy

# Search for a good analogy for Postgres Extensions



can you add on specialized tools  
to Swiss Army Knife?



are extensions similar to Apps  
in an App Store?

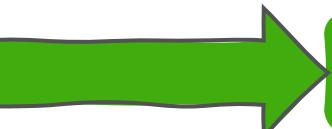
# THE DESIGN OF POSTGRES

*Michael Stonebraker and Lawrence A. Rowe*

*Department of Electrical Engineering  
and Computer Sciences  
University of California  
Berkeley, CA 94720*

## **Abstract**

This paper presents the preliminary design of a new database management system, called POSTGRES, that is the successor to the INGRES relational database system. The main design goals of the new system are to:

- 
- 1) provide better support for complex objects,
  - 2) provide user extendibility for data types, operators and access methods,
  - 3) provide facilities for active databases (i.e., alerters and triggers) and inferencing including forward- and backward-chaining,

“

*We are still dining out on the RADICAL design decision by Michael Stonebraker in the Design of Postgres—to have run-time extensions.”*

—Paul Ramsey, PGConfEU 2018

# Extensions support authored by Dimitri Fontaine in 2011, in Postgres 9.1

[projects](#) / [postgresql.git](#) / commit

[summary](#) | [shortlog](#) | [log](#) | [commit](#) | [commitdiff](#) | [tree](#)  
(parent: [414c5a2](#)) | [patch](#)

commit ▾

Core support for "extensions", which are packages of SQL objects.

committer Tom Lane <tgl@sss.pgh.pa.us>  
Tue, 8 Feb 2011 21:13:22 +0000 (16:13 -0500)  
commit d9572c4e3b474031060189050e14ef384b94e001  
tree 07646762f4086b94a69b9fc215734d2bccade5db  
parent 414c5a2ea65cbd38d79ffdf9b1fde7cc75c134e0

[tree](#)  
[commit](#) | [diff](#)

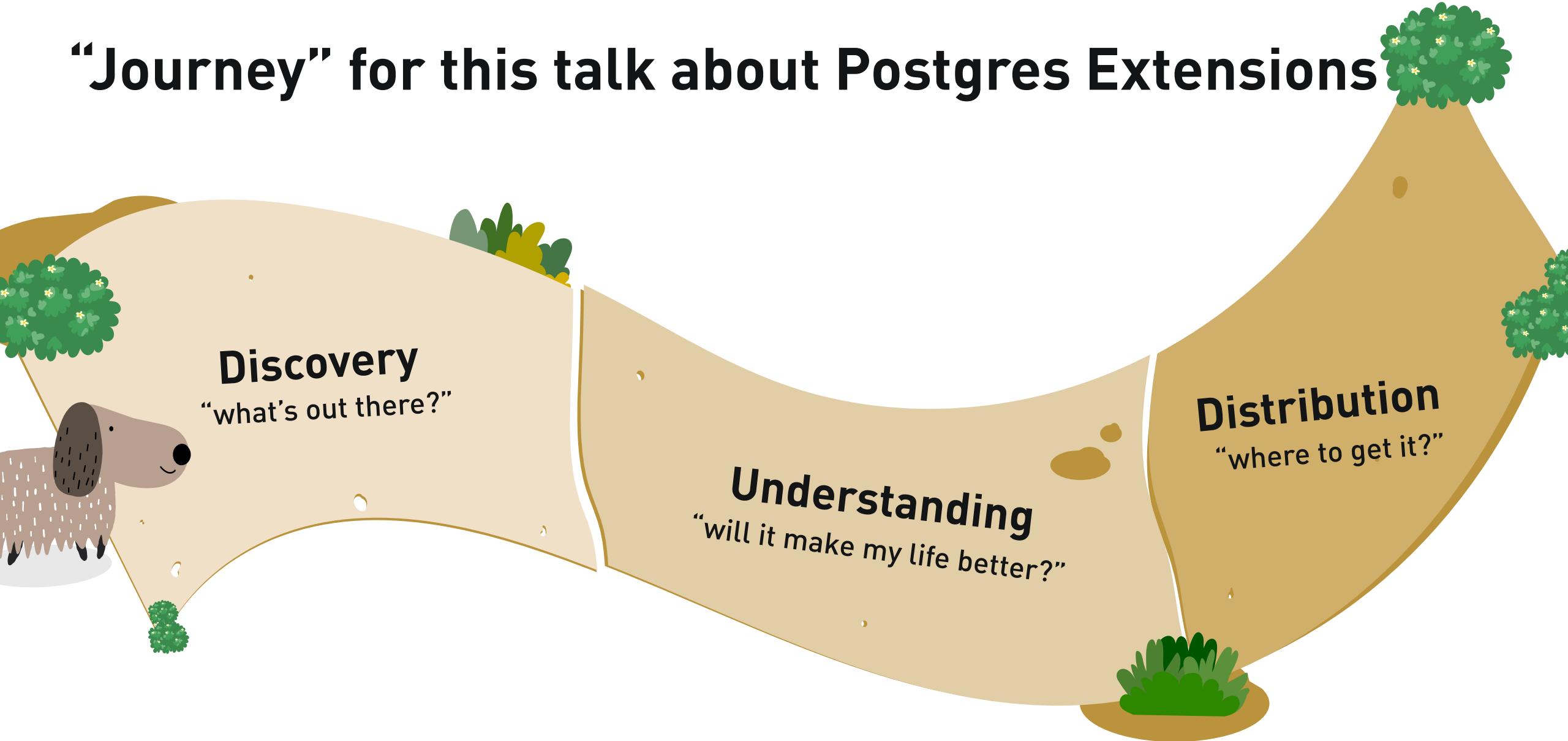
Core support for "extensions", which are packages of SQL objects.

This patch adds the server infrastructure to support extensions.  
There is still one significant loose end, namely how to make it play nice  
with pg\_upgrade, so I am not yet committing the changes that would make  
all the contrib modules depend on this feature.

In passing, fix a disturbingly large amount of breakage in  
AlterObjectNameSpace() and callers.

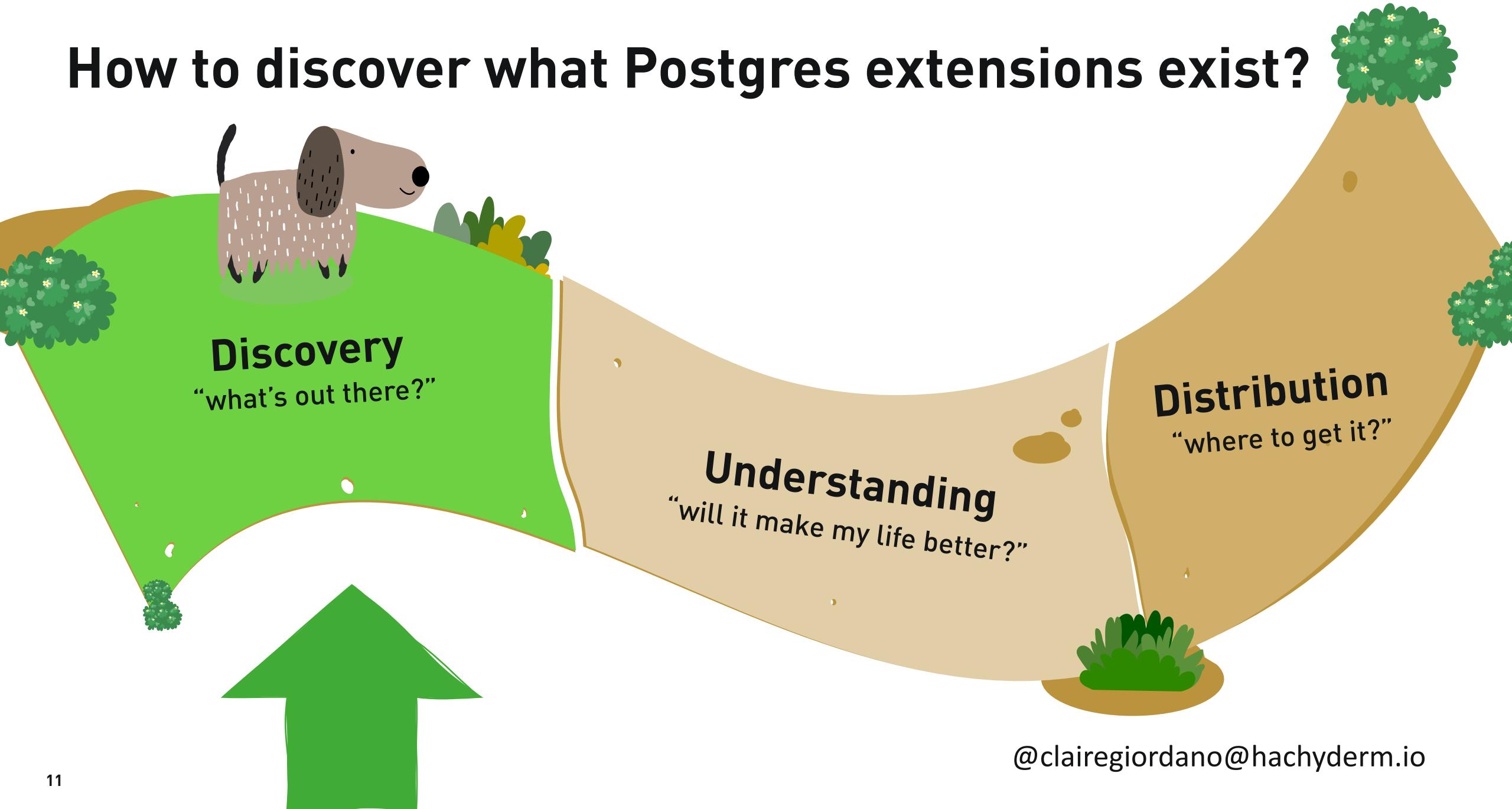
Dimitri Fontaine, reviewed by Anssi Kääriäinen,  
Itagaki Takahiro, Tom Lane, and numerous others

# “Journey” for this talk about Postgres Extensions



@clairegiordano@hachyderm.io

# How to discover what Postgres extensions exist?



# To understand what exists, needed “mental map” of different types of PG extensions

In 1<>1 with Daniel Gustafsson, explored  
this 1<sup>st</sup> draft set of “categories”

1. Monitoring
2. Indexing
3. Data types
4. Geospatial
5. Statistical Analysis
6. Distributed Scale
7. Connectors
8. Storage Optimizations



**But then I started to  
wonder...**

What categories is everyone else using for  
Postgres extensions?

@clairegiordano@hachyderm.io



users tags recent

in Documentation ▾ PGXN Search

acl administration aggregate  
aggregate function analytics arithmetic array arrays audit  
audit log auditing average background worker count  
custom background worker datatype dictionary distinct  
estimate extension external data **fdw**  
**foreign data wrapper**  
function functions hash index  
json jsonb log logging maintenance  
math md5 meta mongodb monitoring parallel  
partitioning performance **plpgsql** postgresql  
python range replication sampling sql med statistics  
**table** template temporal testing trigger type

PGXN, the PostgreSQL Extension network, is a central distribution system for open-source PostgreSQL extension libraries.

#### Recent Releases

##### [db2\\_fdw 6.0.0](#)

PostgreSQL Data Wrappper to DB2 databases

##### [sqlite\\_fdw 2.4.0](#)

Foreign Data Wrapper for SQLite databases

##### [pg\\_extra\\_time 1.0.0](#)

Some date-time functions and operators that, according to the extension author, ought to be part of the PostgreSQL standard distribution.

##### [icu\\_ext 1.8.0](#)

Extension to expose functionality from the ICU (Unicode) library

##### [pg\\_diffix 1.2.0](#)

A PostgreSQL extension for strong dynamic anonymization.

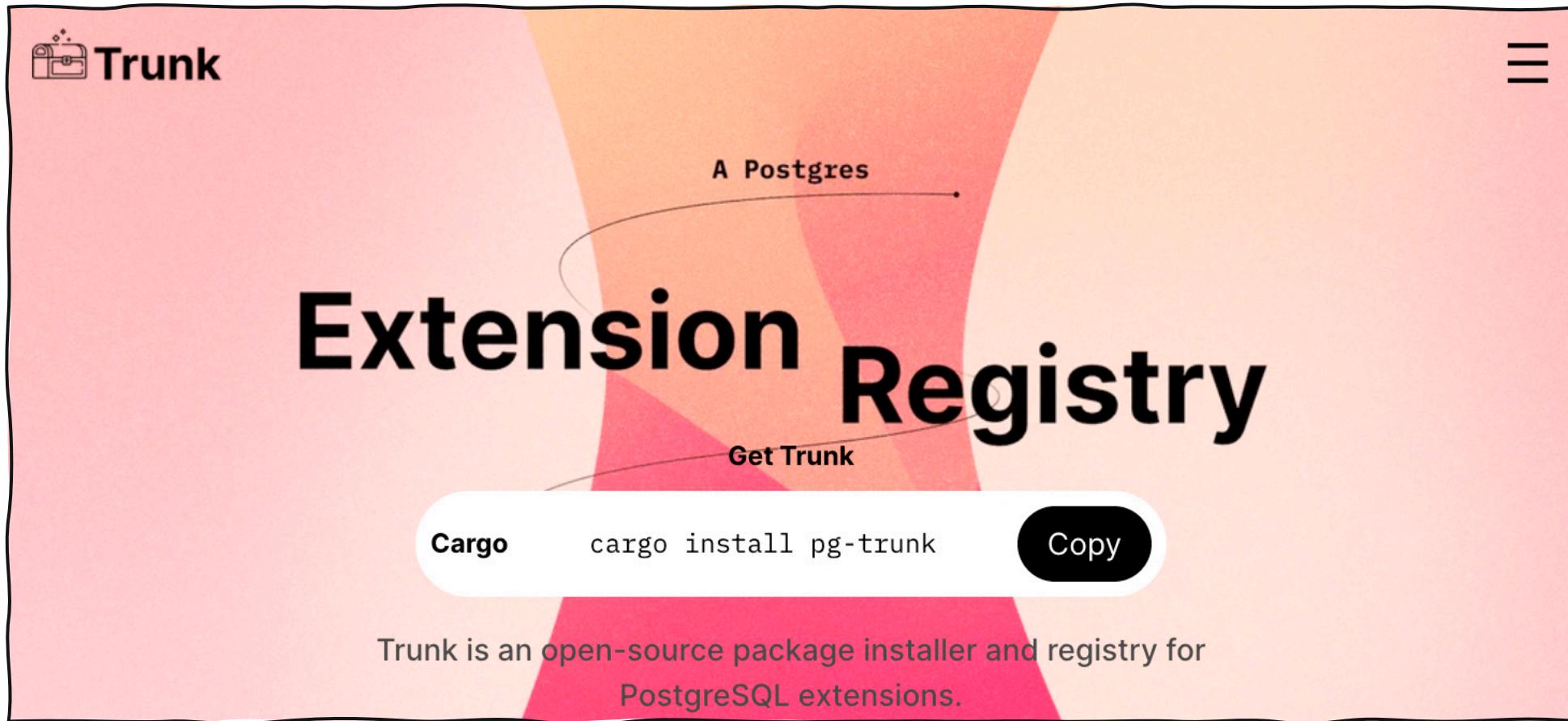
**First place I looked for Categories: PGXN.org**

I found tags!  
Not categories.

I could search!  
Not browse extensions.

# Discovered “Trunk” via Samay Sharma @ Tembo

pgt.dev





# Trunk has categories for extensions!

open source project

pgt.dev

Featured	7	Machine Learning	4
Analytics	12	Metrics	15
Auditing / Logging	7	Orchestration	9
Change Data Capture	6	Procedural Languages	13
Connectors	22	Query Optimizations	5
Data / Transformations	39	Search	8
Debugging	2	Security	10
Index / Table Optimizations	14	Tooling / Admin	13

# Supabase's docs about PG extensions have categories too!

AI

Index

Admin

Language

Audit

Notifications

Cryptography

Search

Data Type

Testing

Dataset

Time Series

Geo

Utility

# So does Crunchy Bridge!



≡

Search docs... ⌘K

Login

Create

Extension	Type of extension	Summary	Command to create
Address Standardizer	Functions	Used to parse an address into constituent elements	<code>CREATE EXTENSION address_standardizer</code>
Address Standardizer (US)	Functions	Data for standardizing US addresses	<code>CREATE EXTENSION address_standardizer</code>
Adminpack	Functions	Administrative functions for Postgres	<code>CREATE EXTENSION adminpack</code>
Amcheck	Functions	Functions for verifying relation integrity	<code>CREATE EXTENSION amcheck</code>
Audit	Functions	Audit user actions	<code>CREATE EXTENSION audit</code>

# Azure Cosmos DB for PostgreSQL has “categories” in docs too

Citus

Data types

Full-text search

Functions

Index types

Language

Miscellaneous

Pgvector

PostGIS

# Like a dog with a tennis ball

I spent wayyyyy too much time  
trying to come up with  
“right” set of categories



# My search to find the “right” set of categories

Claire/Daniel G 1st draft categories	Supabase categories	Trunk categories (from Tembo)	Crunchy Bridge categories	Azure Cosmos DB for PostgreSQL categories	Claire 2nd Take
Monitoring	AI	Analytics	Functions	Distributed Scale	Admin
Indexing	Admin	Auditing/Logging	Logging	Data types	AI / Machine Learning
Data types	Audit	Change Data Capture	Datatype	Full-text search	Analytics
Geospatial	Cryptography	Connectors	Index types	Functions	Connectors
Statistical Analysis	Data Type	Data/Transformations	Views	Index types	Data types
Distributed Scale	Dataset	Debugging	Foreign Data Wrapper	Language	Distributed Scale
Connectors	Geo	Index/Table Optimizations	Dictionaries	Miscellaneous	Geo
Storage Optimizations	Index	Machine Learning	Geospatial utilities	Pgvector	Index
	Language	Metrics		PostGIS	Language
	Notifications	Orchestration			Migrations
	Search	Procedural Languages			Monitoring/Logging
	Testing	Query Optimizations			Performance
	Time Series	Search			Security
	Utility	Security			Text
		Tooling / Admin			



Dimitri Fontaine reminded  
me of this quote



All models are  
wrong, but  
some are useful."

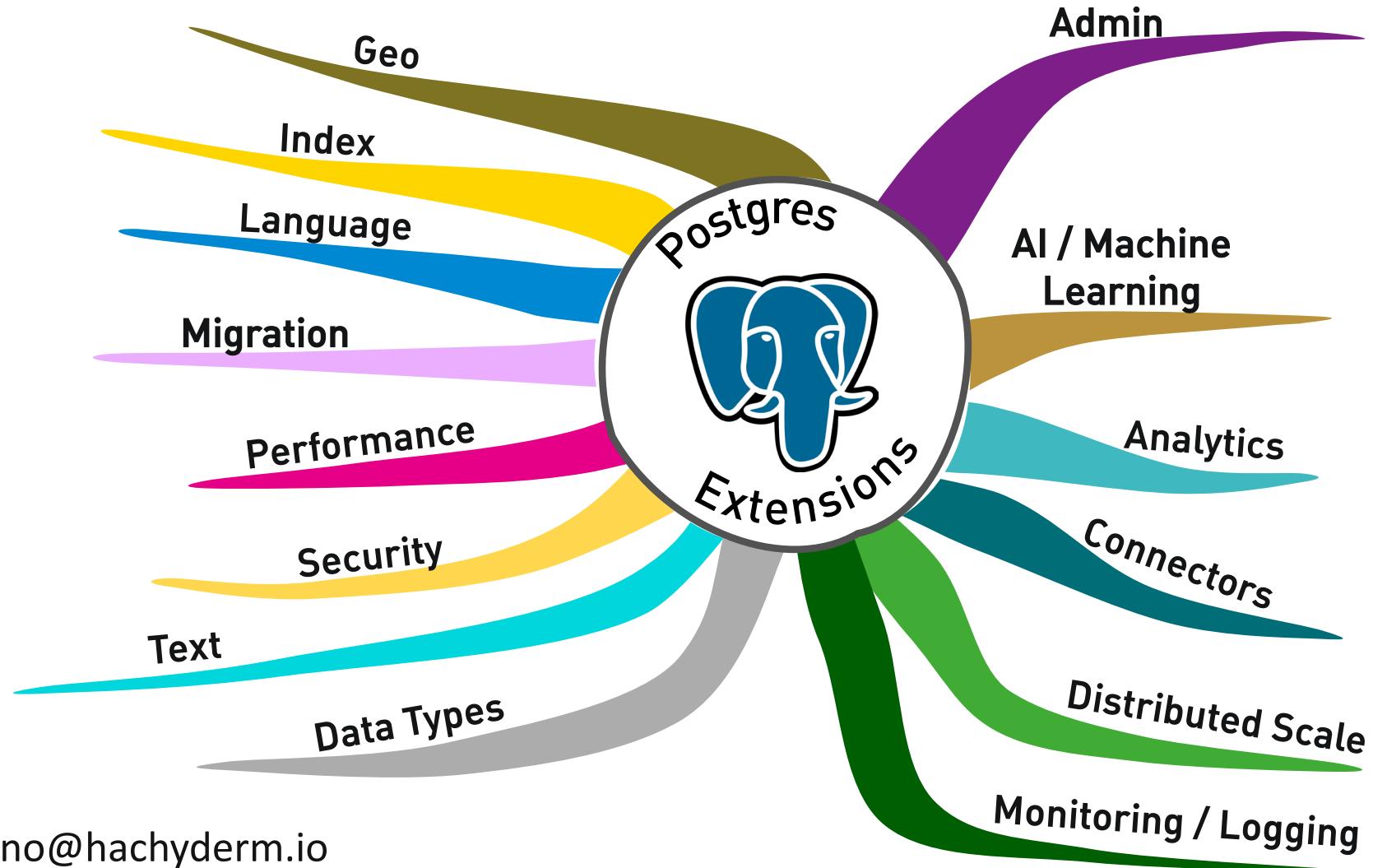
—George Box, statistician

@clairegiordano@hachyderm.io

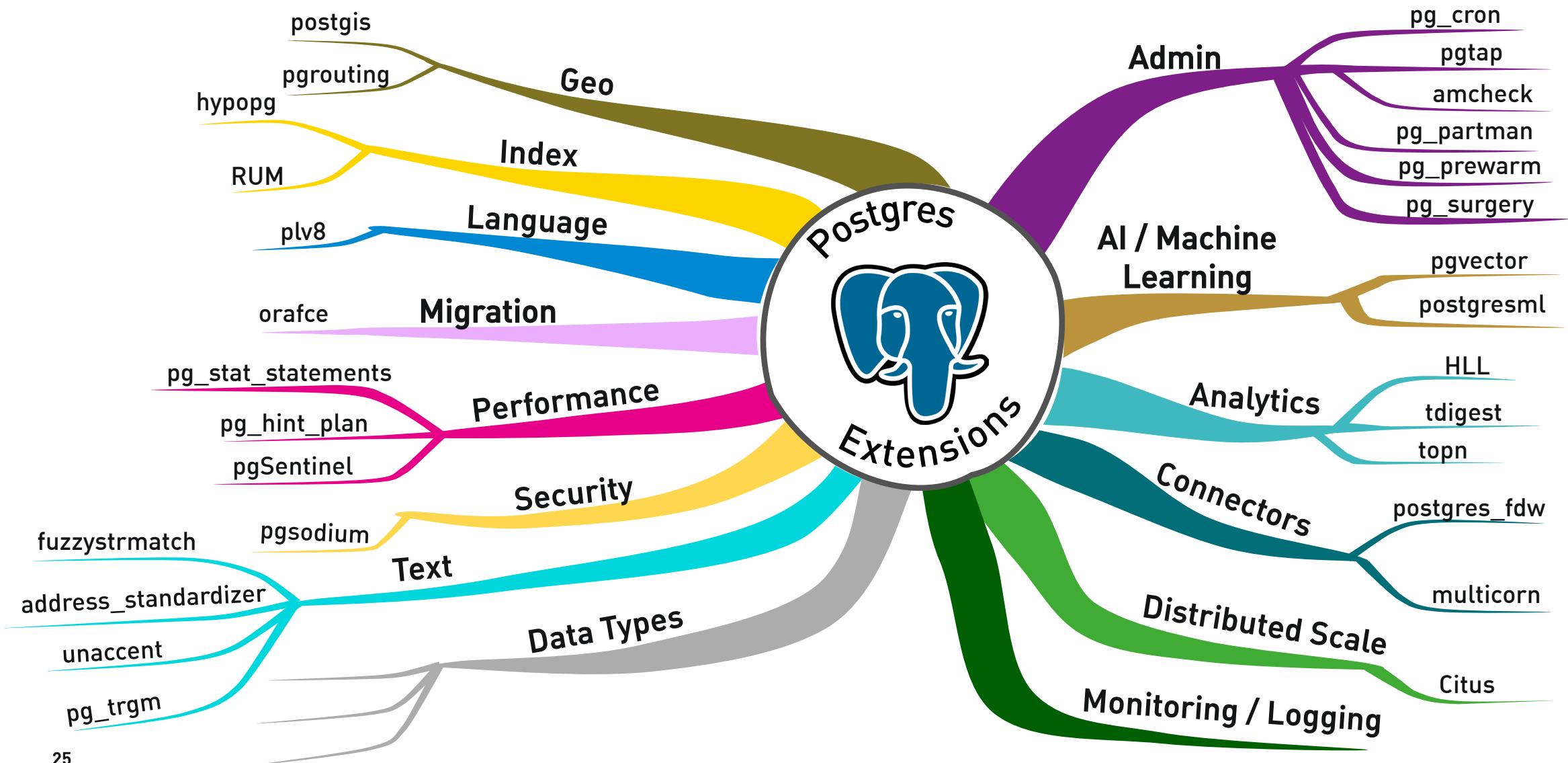
**So I promised you a  
“map”  
of Postgres  
extensions**



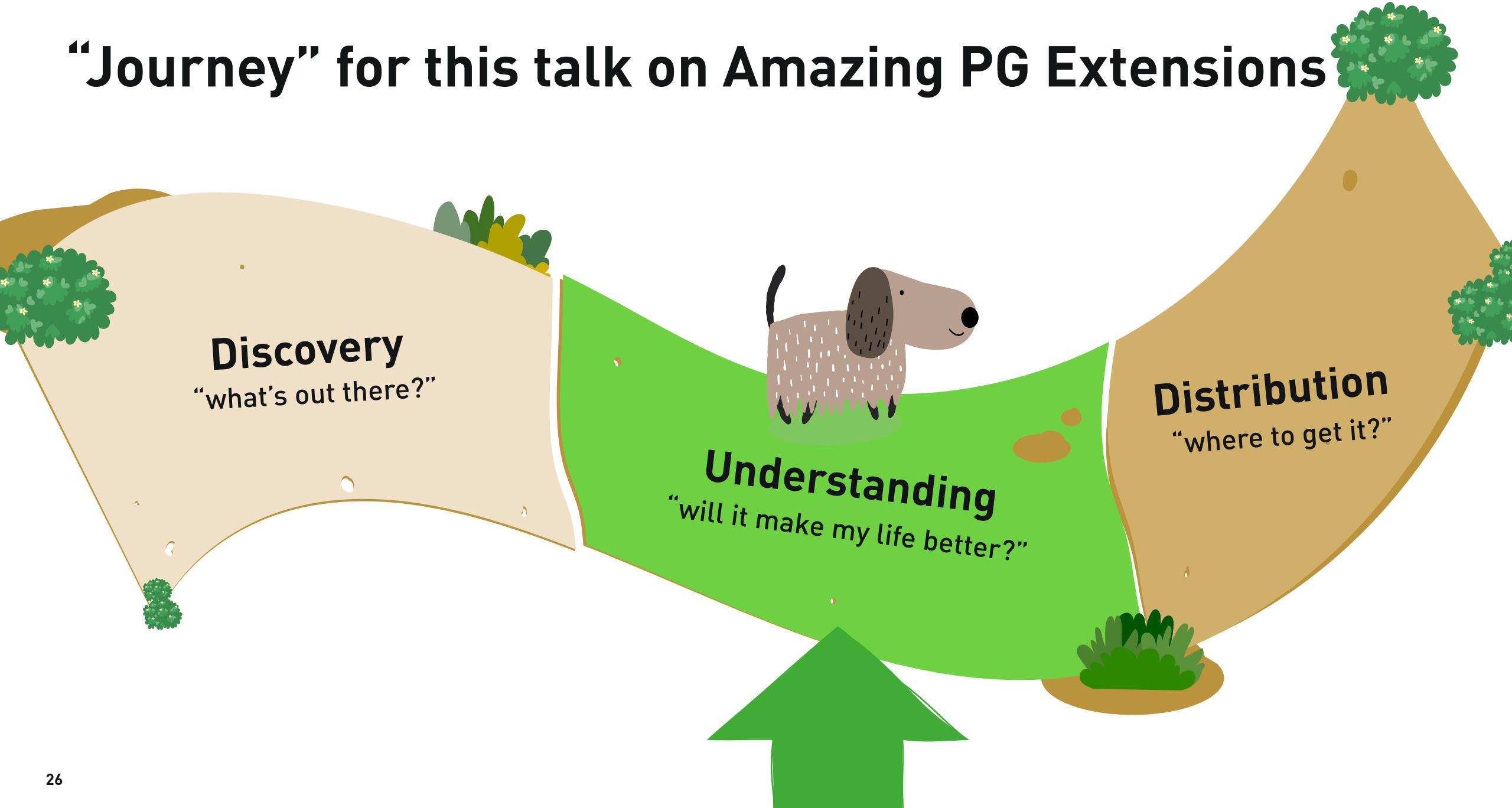
# Mind map of Categories of Postgres Extensions



# Mind map of Postgres Extensions (by category)



# “Journey” for this talk on Amazing PG Extensions



And will Postgres extensions save you time?

30K DAYS

CATEGORY

Performance

# pg\_stat\_statements

CONTRIB

tracks statistics about SQL statements

how long queries took?



Claire Giordano ✨ @clairegiordano@hachyderm.io  
@clairegiordano · Follow



Most frequently dispensed #PostgreSQL tip-of-the-day here in the Postgres devroom at #FOSDEM? Use pg\_stat\_statements! @Xof's talk on Breaking PostgreSQL at Scale is the 4th talk today to drive this point home HT @craig @net\_snow @magnushagander



5:31 AM · Feb 3, 2019



28 Reply Copy link

CATEGORY

Connectors

postgres\_fdw

mongo\_fdw

mysql\_fdw

duckdb\_fdw

tds\_fdw

oracle\_fdw

kafka\_fdw

# postgres\_fdw

CONTRIB

fdw = “Foreign Data Wrapper”

lots and lots of foreign data wrappers  
(just a few listed here)

# Concept of “Super extension” on Postgres.FM

The screenshot shows the Postgres.FM website with a white header and a teal background. In the top-left corner, there's a circular icon containing a microphone. The header includes the logo 'postgres.fm' in pink, and navigation links 'HOME', 'EPISODES', and 'SUBSCRIBE'. Below the header, a black box displays the date 'JUNE 9, 2023 • 30 MINUTES'. The main content area features the title 'Extensions' in large bold letters. Underneath it are two buttons: 'PLAY EPISODE' and 'DOWNLOAD (27.8 MB)'. To the right of the title is a pink rectangular image with the 'postgres.fm' logo and a cartoon elephant wearing sunglasses. At the bottom, there's a section titled 'Show Notes / Transcript' with a descriptive paragraph about Postgres extensions.

**postgres.fm**

HOME    EPISODES    SUBSCRIBE

JUNE 9, 2023 • 30 MINUTES

## Extensions

**PLAY EPISODE**    **DOWNLOAD (27.8 MB)**

**Show Notes / Transcript**

Nikolay and Michael discuss Postgres extensions — what they are, how they affect your decisions around Postgres, and some things to keep in mind when using them.

CATEGORY

Geo

# PostGIS

store, index, & query  
geographic data

points, lines, polygons

“Super extension”

# PostGIS



Podcast episode with  
Paul Ramsey & Regina Obe:  
“Why people care about PostGIS”



**PATH TO  
CITUS CON**

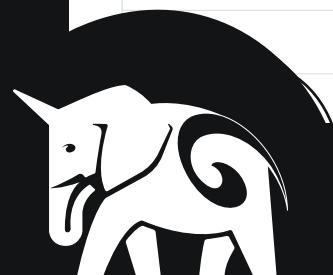
For Developers Who Love Postgres

CATEGORY

Distributed Scale

The GitHub repository page for `citusdata / citus` displays the following information:

- Code**: 6,718 commits, 598 branches, 144 tags.
- Issues**: 897
- Pull requests**: 65
- Discussions**, **Actions**, **Wiki**, **Security**, **Insights**
- Repository Details**: citus (Public), Edit Pins, Watch 212, Fork 596, Starred.
- Commits** (Recent):
  - onerkalaci Addd missing image to ... (yesterday)
  - .circleci Bump cryptography from 41.0.3 to 41.0.4 in ... (4 days ago)
  - .github Removes centos 7 for PG 16 in packaging pi... (2 weeks ago)
  - ci Fix string eval bug in migration files check (... (7 months ago)
  - config Renames configure.in to fix warnings (#6034) (last year)
  - images Addd missing image to Technical Readme (...) (yesterday)
  - Addd missing image to Technical Readme (...) (yesterday)
  - Update cherry-pick hash in vendor README (3 years ago)
- About**: Distributed PostgreSQL as an ex... [www.citusdata.com](http://www.citusdata.com)
- Tags**: postgres, multi-tenant, sql, d..., scale, postgresql, sharding, distributed-database, citus, relational-database, database-clus, citus-extension
- Links**: Readme, AGPL-3.0 license, Code of conduct



# Citus

Distributed PostgreSQL

“Postgres at any scale”

Sharding

Led by Microsoft

“Super extension”

# Postgres extensions enable innovation

Robert Haas @robertmhaas · Jun 5

Maybe [@PostgreSQL](#) needs TIMESTAMP WITH STORED TIME ZONE  
(timestampstz)

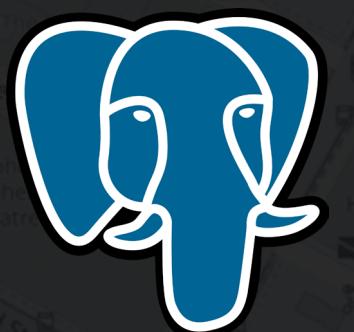
5 18 1,356

xzilla @robtreat2

“There’s an extension for that” TM

pgxn.org

timestampandtz: Timestamp and timezone data type for ...  
timestampandtz is a data type for PostgreSQL that stores the timestamp and the source timezone



# LESSER KNOWN AND FAVORITE EXTENSIONS

CATEGORY

Geo

# pgRouting

adds routing  
network analysis

Shortest Path

Driving Distance

Traveling Salesperson

## pgRouting: A Practical Guide

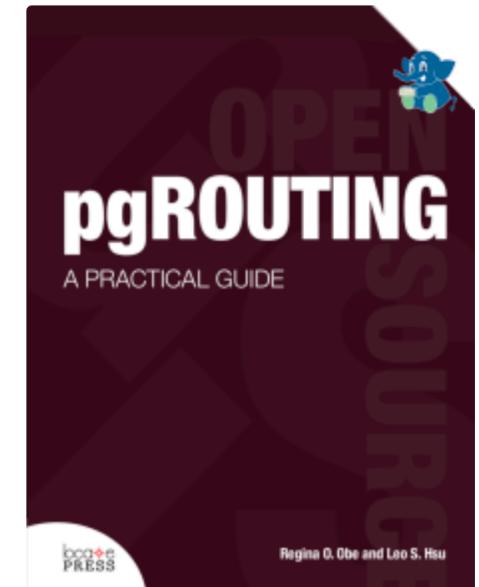
by Regina Obe and Leo Hsu

What is pgRouting? It's a PostgreSQL extension for developing network routing applications and doing graph analysis.

Interested in pgRouting? If so, chances are you already use PostGIS, the spatial extender for the PostgreSQL database management system. So when you've got PostGIS, why do you need pgRouting? PostGIS is a great tool for molding geometries and doing proximity analysis, however it falls short when your proximity analysis involves constrained paths such as driving along a road or biking along defined paths.

PostGIS alone can't help you apply costs and resource constraints to your travels, such as traffic congestion, mileage restrictions, or allocating resources like vehicles. pgRouting is a complement to PostGIS that allows you to incorporate costs and path restrictions to your proximity analysis.

pgRouting is well suited to building travel navigation systems and computing driving distance polygons for places of interest. While the main focus of pgRouting is GIS applications, it can be



[Buy Paperback](#) [Buy PDF](#) [Amazon](#)

PDF \$34.99

282 pages

ISBN 978-0989421737

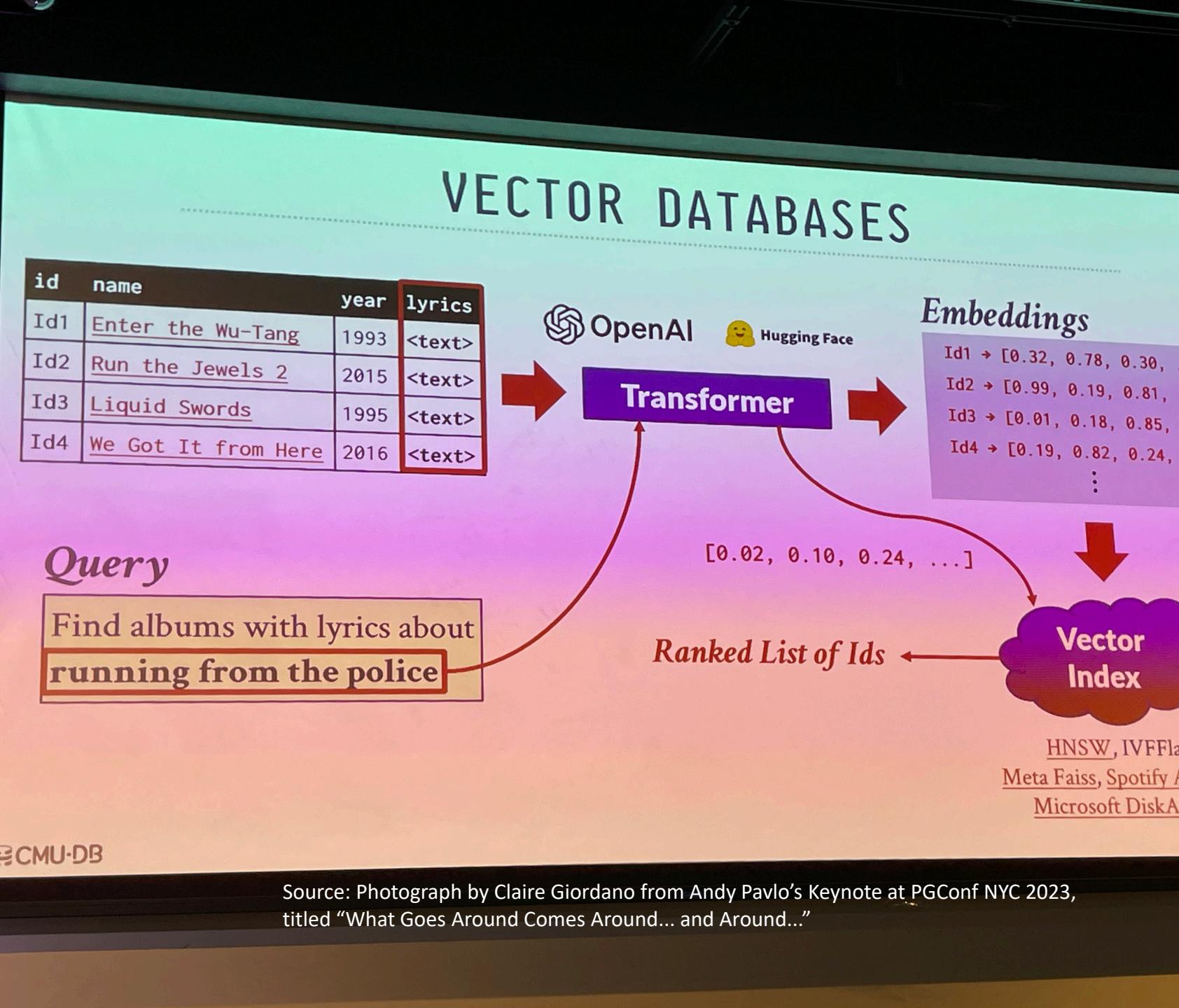
Published 2017-05-31

# pgvector

open source  
vector similarity search

has added HNSW as new  
index type

HNSW = Hierarchical  
Navigable Small World  
(HNSW)



Source: Photograph by Claire Giordano from Andy Pavlo's Keynote at PGConf NYC 2023,  
titled "What Goes Around Comes Around... and Around..."

CATEGORY

Admin

# pg\_cron

Created/maintained by  
Marco Slot

- - vacuum my table every night at 3am (GMT)

```
SELECT cron.schedule( '0 3 * * *',
'VACUUM my_table' );
```

- - call a procedure every minute

```
SELECT cron.schedule('process-new-events',
'* * * * *', 'CALL process_new_events()');
```

**“pg\_cron: probably the best way to schedule jobs within PostgreSQL database”**

—FatDBA

## Efficient approximate analytics functions

### t-digest

Efficient percentiles for parallel apps / OLAP index.

```
INSERT INTO t SELECT random() FROM generate_series(1,10000);
SELECT tdigest_percentile(c, 100, 0.95) FROM t;
tdigest_percentile
-----
0.9513706724006539
(1 row)
```

### hll

Estimate distinct values with high accuracy.

```
INSERT INTO hll_example (hll_set)
SELECT
hll_add_agg(hll_hash_integer(i)) FROM generate_series(1, 1000) i;
SELECT hll_cardinality(hll_set) FROM hll_example;
hll_cardinality
-----
9725.852733707077
(1 row)
```

### topN

Set rule-based top value returns.

```
INSERT INTO events VALUES ('a'), ('b'), ('a'), ('a');
SELECT (topn(topn_add_agg(name), 2)).* FROM events GROUP BY name;
item | frequency
-----+
a    |      3
b    |      2
(2 rows)
```

CATEGORY

Index

# tdigest, hll, topn

approximation/"sketch" algorithms

t-digest: calculate Percentiles (histograms)

hll = HyperLogLog, for COUNT DISTINCT

CATEGORY

Index

# RUM

more powerful version of  
GIN index for text search  
(or your own custom  
types)

stores additional info per  
entry, e.g. timestamp



CATEGORY

Index

# RUM

more powerful version of  
GIN index for text search  
(or your own custom  
types)

supabase DOCS

Database

## RUM: improved inverted index for full-text search based on GIN index

RUM is an extension which adds a RUM index to Postgresql.

RUM index is based on GIN that stores additional per-entry information in a posting tree. For example, positional information of lexemes or timestamps. In comparison to GIN it can use this information to make faster index-only scans for:

- Phrase search
- Text search with ranking by text distance operator
- Text `SELECT` s with ordering by some non-indexed additional column e.g. by timestamp.

HypoPG / hypopg

Code Issues 1 Pull requests 1 Actions Projects Wiki Security ...

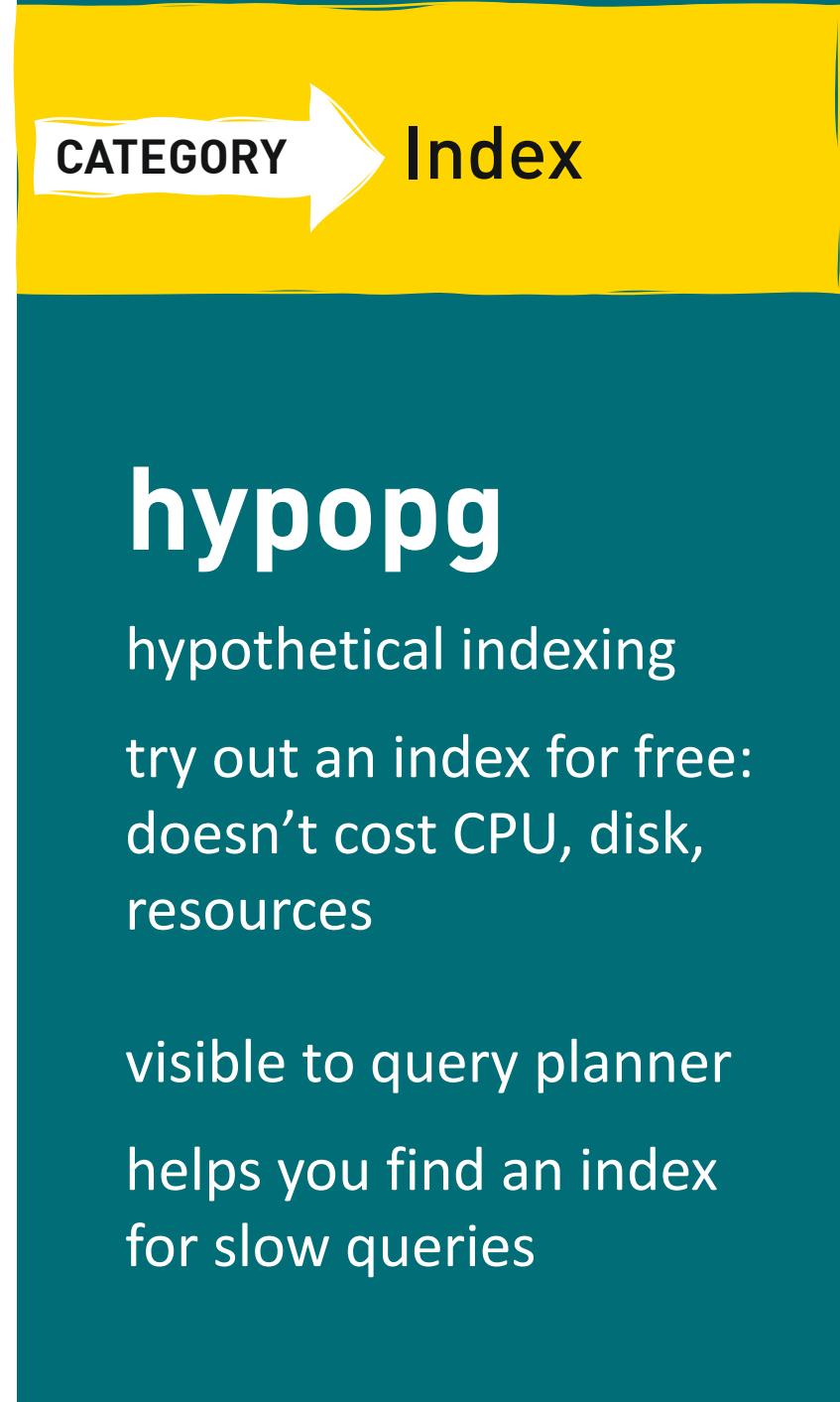
Preview Code Blame 186 lines (142 loc) · 7.37 KB Raw ⌂ ⌄ ⌅ ⌆

## HypoPG [🔗](#)

HypoPG is a PostgreSQL extension adding support for hypothetical indexes.

An hypothetical -- or virtual -- index is an index that doesn't really exists, and thus doesn't cost CPU, disk or any resource to create. They're useful to know if specific indexes can increase performance for problematic queries, since you can know if PostgreSQL will use these indexes or not without having to spend resources to create them.

For more thorough informations, please consult the [official documentation](#).



CATEGORY

Performance

# pg\_hint\_plan



It's an excellent learning tool, really good for deep dive performance work. Why? Because it gives you ability to pick and choose between different indexes—and get the execution plan without having to rebuild those indexes... The larger the data set the more valuable it is."



—Rob Treat, statistician

CATEGORY

Performance

# `pgsentinel` – sampling active session history

build unknown

## Introduction

PostgreSQL provides session activity. However, in order to gather activity behavior user have to sample the `pg_stat_activity` view multiple times.

`pgsentinel` is an extension to record active session history and also link the activity with query statistics ( `pg_stat_statements` ).

# `pgsentinel`

a way to get what's going on in the database over time

you can look at “right now” with `pg_stat_activity`, but maybe something important happened 5 minutes ago?

or maybe you need statistics about “what's been happening in my database”

created/maintained by Bertrand Drouvot

CATEGORY

Language

plv8

procedural language

lets you run JavaScript  
code

stored procedures,  
triggers

## plv8/plv8

V8 Engine Javascript Procedural Language add-on  
for PostgreSQL



37  
Contributors

3  
Issues

2k  
Stars

130  
Forks



**GitHub - plv8/plv8: V8 Engine Javascript Procedural Language  
add-on for PostgreSQL**

[github.com](https://github.com)

CATEGORY

Language

plv8

procedural language

lets you run JavaScript  
code

stored procedures,  
triggers

## plv8/plv8

V8 Engine Javascript Procedural Language add-on  
for PostgreSQL



37  
Contributors

3  
Issues

2k  
Stars

130  
Forks

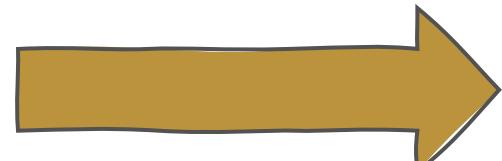


GitHub - plv8/plv8: V8 Engine Javascript Procedural Language  
add-on for PostgreSQL

[github.com](https://github.com)

NEXT TALK:

“Building a better JavaScript  
Postgres Language Extension”  
by Jerry Sievert





CATEGORY

Migration

# orafce

makes it easier to  
migrate from Oracle to  
Postgres

functions, data types...

CATEGORY

Admin

# pgTAP



The TAP in pgTAP stands for “test anything protocol.” pgTAP enables you to write unit tests for your database—for objects, tables, & views. Developers will tell you it’s really really cool.”

—Daniel Gustafsson,  
Postgres hacker at Microsoft





# pg\_prewarm

## CONTRIB

useful to put pages back  
in cache after restarting  
Postgres

maybe should have been  
in “Performance”  
category?

CATEGORY

Admin

# pg\_surgery

to recover from  
corruption

beware!  
“unsafe by design”





CATEGORY

Admin

amcheck

CONTRIB

checks internal  
consistency

tables & indexes

CATEGORY → Admin

pg\_partman





CATEGORY

Security

pgsodium

CATEGORY

Text

# Unaccent

```
1 CREATE EXTENSION unaccent;
2 SET client_encoding TO 'UTF8';
3 SELECT unaccent('Álvaro Mejías');
4 | unaccent
5 -----
6 | Alvaro Mejias
7 (1 row)
```



# postgres

`"pos", "ost", "stg", "tgr",  
"gre", "res"`

Source: pganalyze blog post: Similarity in Postgres and Rails using Trigrams

CATEGORY

Text

## fuzzystrmatch

CONTRIB

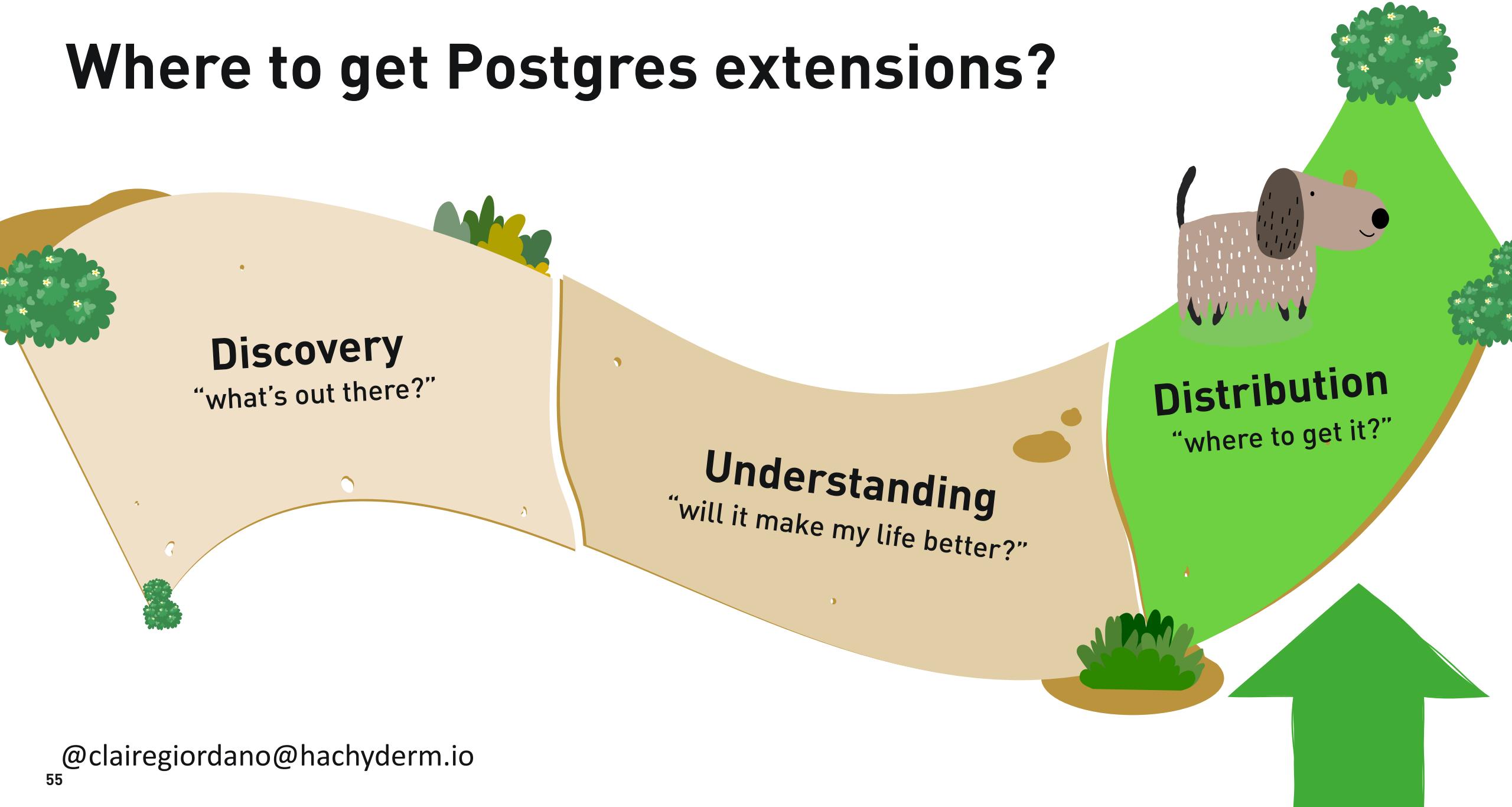
“fuzzy string match”

## pg\_trgm

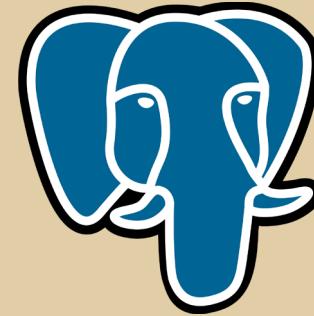
CONTRIB

“trigram”

# Where to get Postgres extensions?



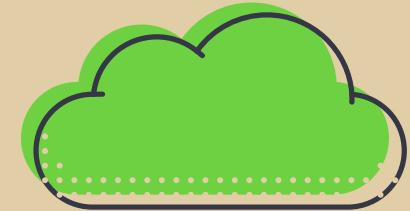
# 4 common sources for Postgres extensions...



“Contrib”  
(pre-bundled)

RPM &  
Debian

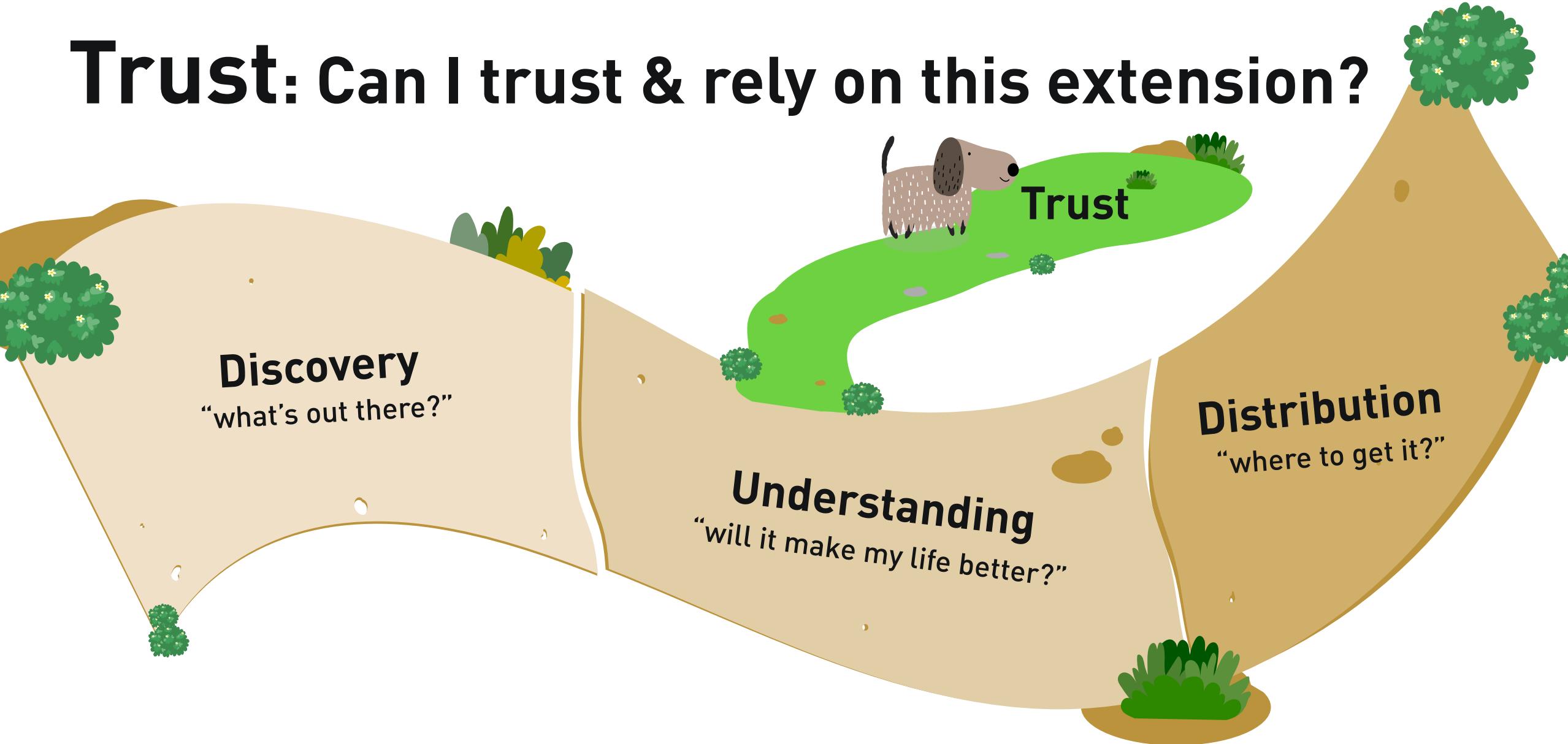
**PGXN**  
PostgreSQL Extension Network



Cloud provider  
(if managed  
service)

+1 to Christoph Berg, Devrim Gündüz, David Wheeler, & all PG extension creators!

# Trust: Can I trust & rely on this extension?



@clairegiordano@hachyderm.io

# Reflections on Trusting Trust

*To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.*

KEN THOMPSON

## INTRODUCTION

I thank the ACM for this award. I can't help but feel that I am receiving this honor for timing and serendipity as much as technical merit. UNIX<sup>1</sup> swept into popularity with an industry-wide change from central mainframes to autonomous minis. I suspect that Daniel Bobrow [1] would be here instead of me if he could not afford a PDP-10 and had had to "settle" for a PDP-11. Moreover, the current state of UNIX is the result of the labors of a large number of people.

There is an old adage, "Dance with the one that brought you," which means that I should talk about UNIX. I have not worked on mainstream UNIX in many

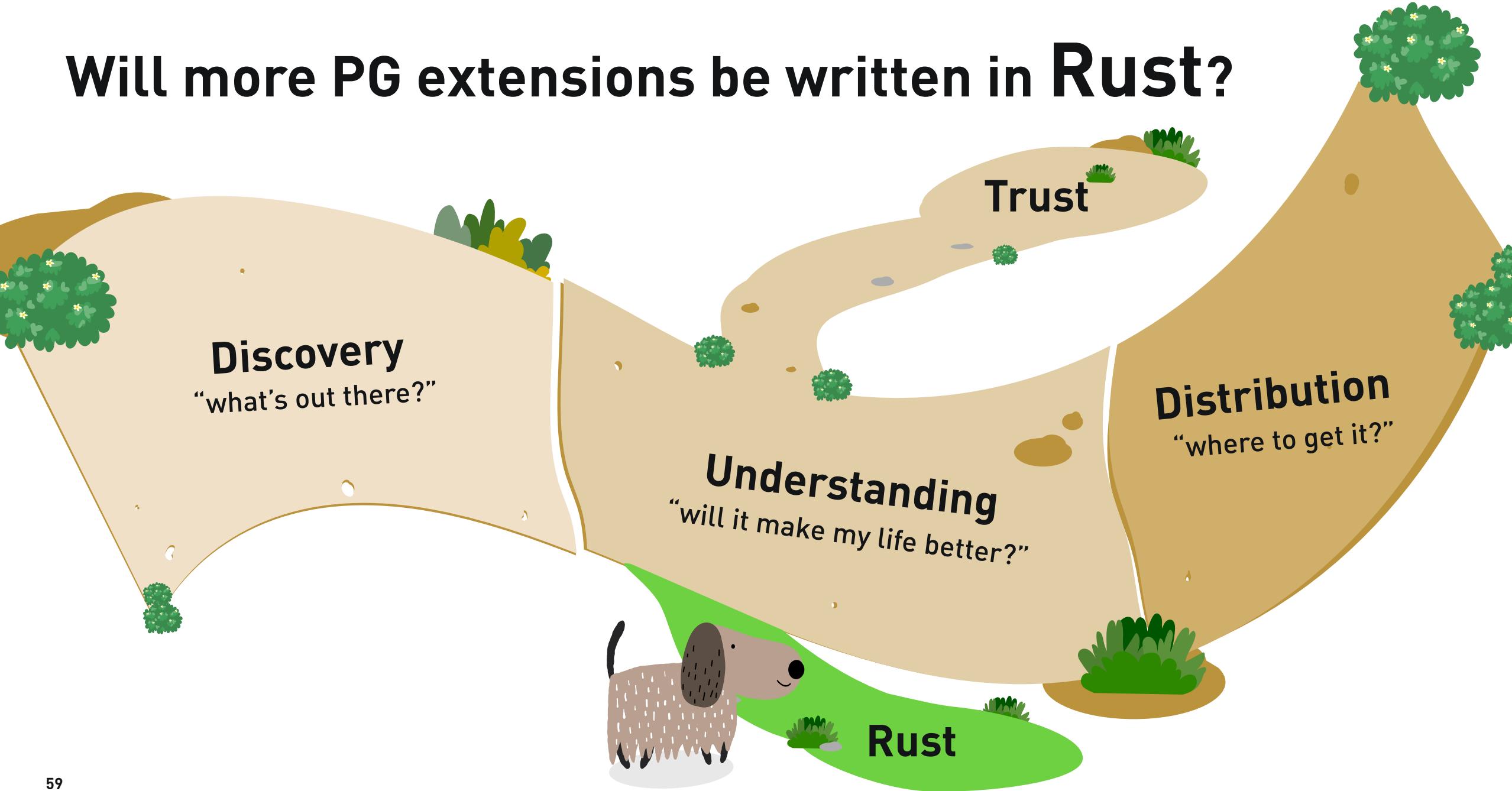
programs. I would like to present to you the cutest program I ever wrote. I will do this in three stages and try to bring it together at the end.

## STAGE I

In college, before video games, we would amuse ourselves by posing programming exercises. One of the favorites was to write the shortest self-reproducing program. Since this is an exercise divorced from reality, the usual vehicle was FORTRAN. Actually, FORTRAN was the language of choice for the same reason that three-legged races are popular.

Source: cs.cmu.edu. Copyright 1984 ACM. Originally published in Communications of the ACM, August 1984, Volume 27, Number 8. Copy by permission of the Association of Computing Machinery.

# Will more PG extensions be written in Rust?



# PGRX = Platform for building Rust extensions

Created by Eric Ridge of  
ZomboDB...

Used by 62 repos so far  
2.7K stars

pgcentralfoundation / pgrx

Code Issues 140 Pull requests 16 Actions Security Insights

pgrx Public Sponsor Watch 35 Fork 179 Starred 2.7k

develop Go to file Add file Code

Branches Tags

workingjubilee Make check... 4 days ago 1,682

.cargo Use .cargo/config.toml... last year

.github Float Dockerfile's Fedora v... 2 weeks ago

art PgCentral Foundation upd... 3 months ago

articles PgCentral Foundation upd... 3 months ago

cargo-pgrx Fix install command for P... 2 weeks ago

nix v0.10-develop branch up... 3 weeks ago

pgrx-exa... Assume commutation wh... 2 months ago

pgrx-mac... v0.10-develop branch up... 3 weeks ago

pgrx-pg... v0.10-develop branch up... 3 weeks ago

About

Build Postgres Extensions with Rust!

rust postgres postgresql  
rustlang postgresql-extension

Readme View license Security policy Activity

2.7k stars 35 watching 179 forks Report repository

Releases 96

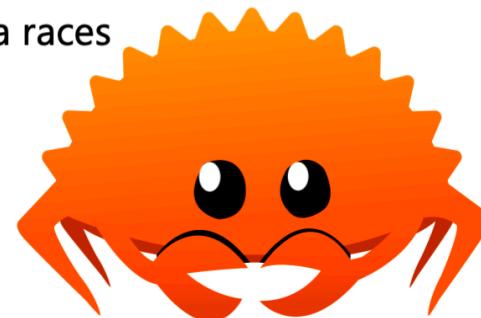
# Not only do developers love Rust—extensions that are built in Rust will benefit users too



I like Rust



- It feels modern: everything is an expression + pattern matching
- Its type system is fancy: Like Haskell but I can read it
- Secure: No memory leaks, no double frees, no data races
- It's as fast as C/C++
- It has a package manager



Rust logo by Rust Foundation, CC BY 4.0, <https://commons.wikimedia.org/w/index.php?curid=40715219>

Source: Jelte Fennema talk “[Postgres without SQL: Natural language queries using GPT-3 & Rust](#)” at Citus Con: An Event for Postgres 2023. Published on YouTube.

# People to thank for inspiration & reviews



Daniel Gustafsson  
Devrim Gündüz  
Dimitri Fontaine  
Jelte Fennema-Nio  
Marco Slot  
Melanie Plageman

Nik Larin  
Robert Treat  
Samay Sharma  
Teresa Giacomini  
Thomas Munro



# Thank you

@clairegiordano@hachyderm.io



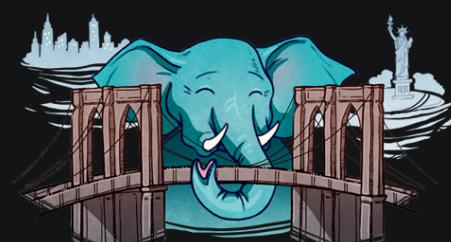
| @clairegiordano



aka.ms/PathToCitusCon



PGCONF NYC 2023



Microsoft