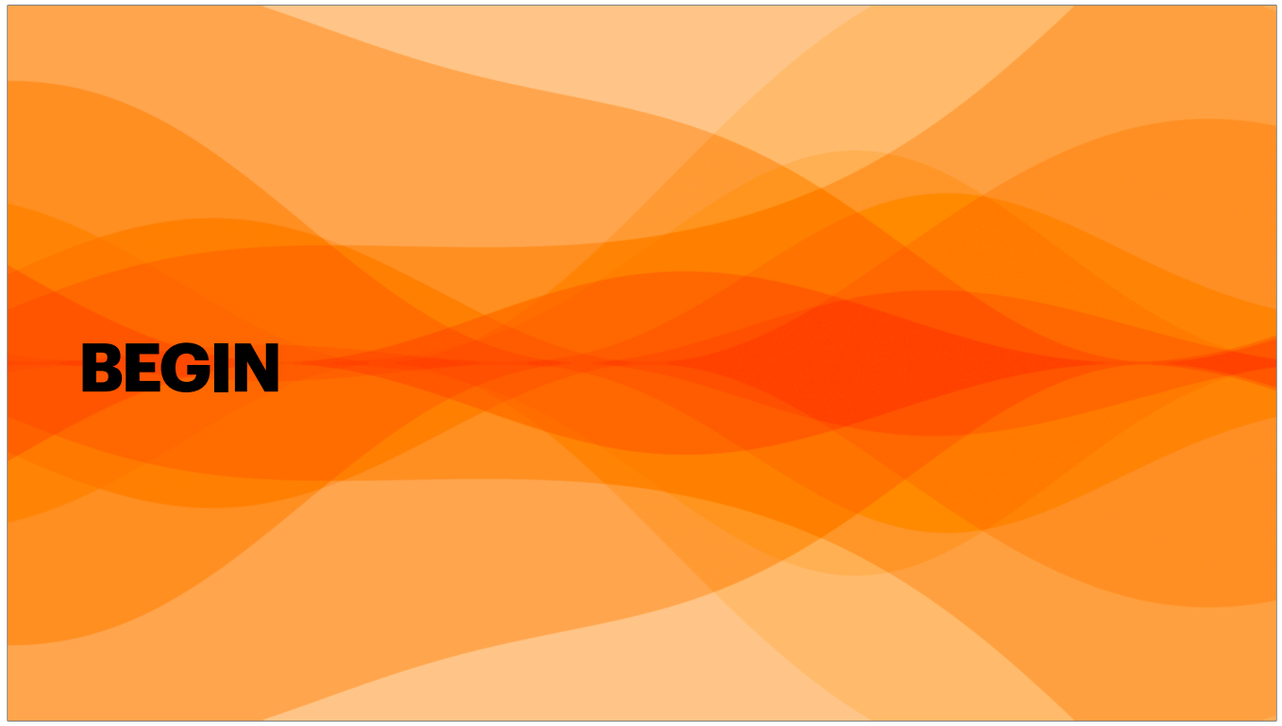

G’day everyone, and welcome to the first Postgres Extension Ecosystem Summit EU. Thanks for coming! I’m David Wheeler, creator of PGXN and Principal Architect at Tembo.

I’d like to thank Floor Drees for organizing this event, where we’ve got some great extensions to work on today. But first, I’d like to talk about the state of the Postgres Extension Ecosystem, its past, present, and future.




Let’s go back in time, to the beginning.

# LEGACY

Long history of extensibility

Two approaches

- shared_preload_libraries
- Pure SQL (including PLs)

A few intrepid adopters

- PostGIS
- BioPostgres
- PL/R
- PL/Proxy
- pgTAP

- Postgres has a long history of extensibility
- In the old days, there were two basic approaches to extending Postgres without forking it
    - 1.Load dynamic shared objects (DSOs) via the shared_preload_libraries GUC;
    - 2. Execute SQL and procedural languages to create custom functions and other objects
- There were quite a few intrepid adopters in those years
    - Including PostGIS, BioPostgres, PL/R, PL/Proxy, pgTAP, and others

# POSTGRES 9.1

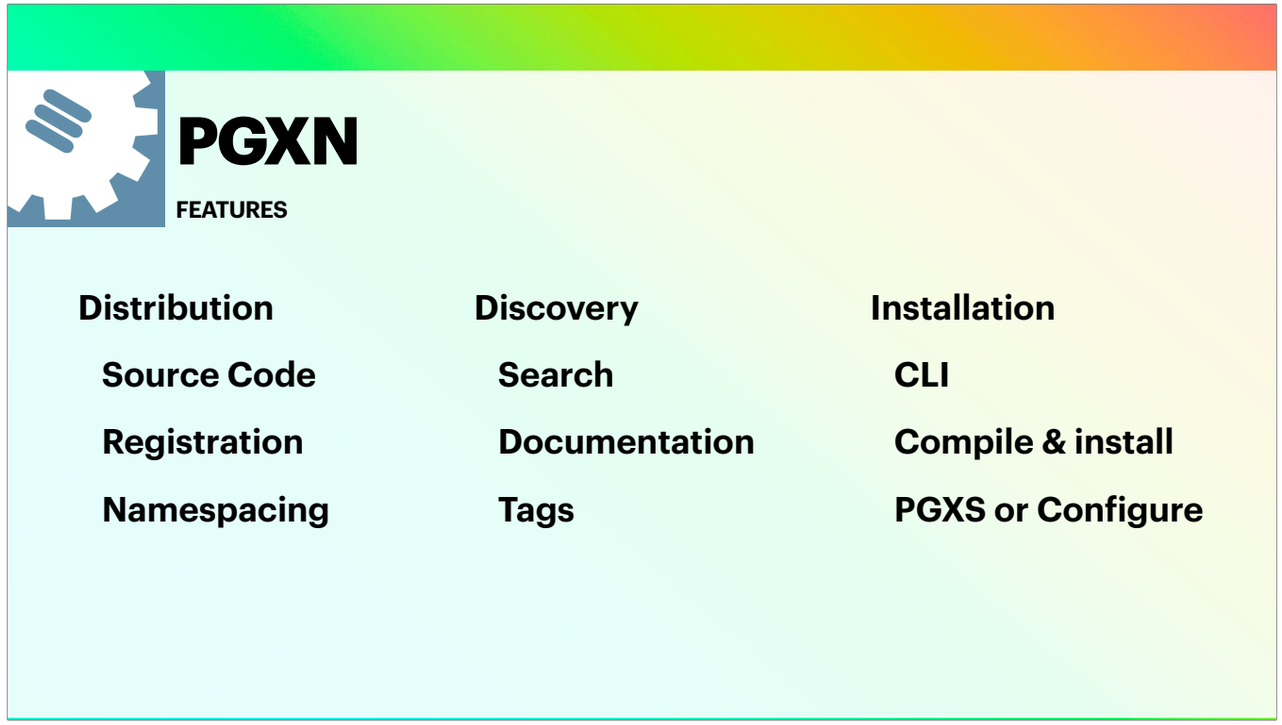
**Dimitri Fontaine adds extensions**

**Key features:**

- **Compile and install**
- **CREATE/UPDATE/DROP EXTENSION**
- **pg_dump and pg_restore**

- In 2011, Dimitri Fontaine’s patch for formal extension support was released in Postgres 9.1.
- This work built on those existing patterns, but formalized things through three key features:
  - Tooling to compile and install extensions, both DSOs and pure SQL
  - New SQL commands to create, update, and drop extensions as named units
  - And backup and restore support to ensure consistent versioning and behavior when upgrading Postgres itself


- All this infrastructure created new opportunities for the extensibility of Postgres.




- Around the time Dimitri started developing his patch, I proposed to create PGXN, a service for distributing Postgres extensions, with the goal to be the canonical source for all publicly-available extensions
- The features were to include distribution of source code, user registration, and namespace management
- Comprehensive search features, and the ability to browse and read well-written documentation.
- And a simple CLI as to download, compile, and install extensions.

# PGXN

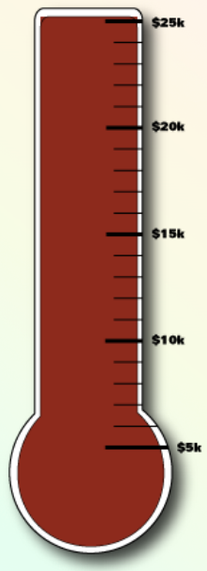
HISTORY

Funded December 2010

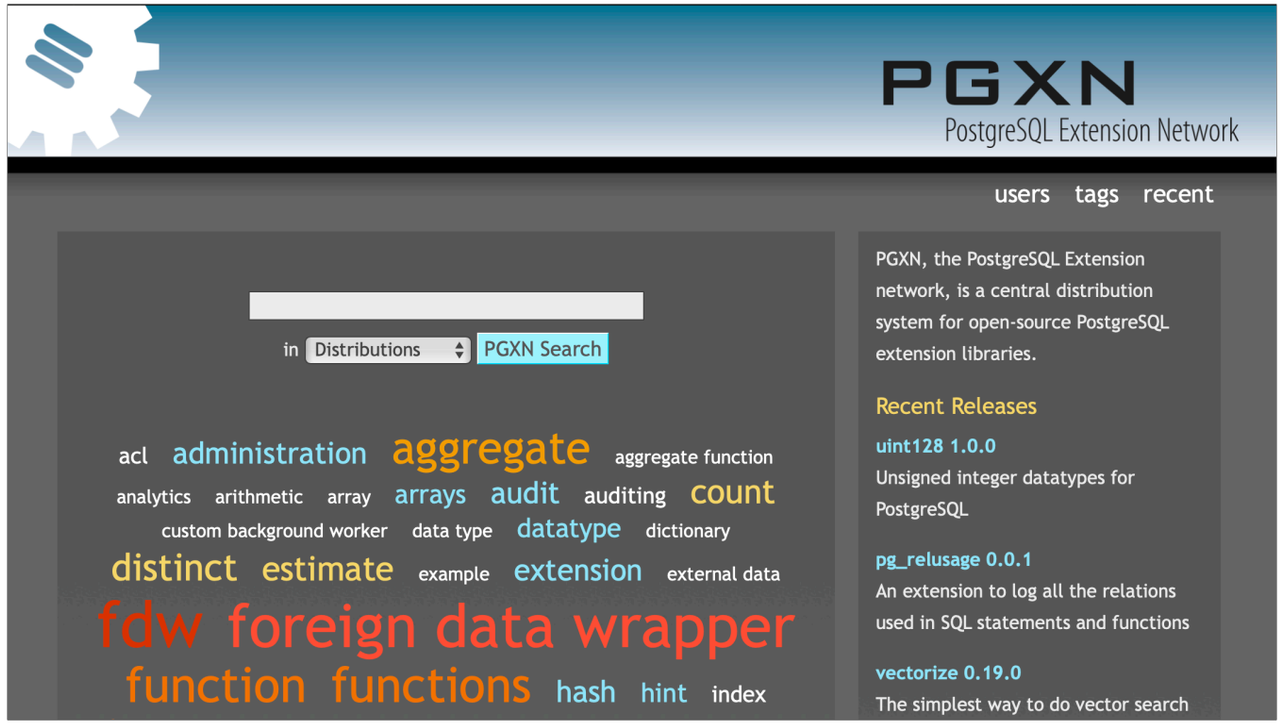
pgxn.org launched April 2011

Client by  Daniele Varrazzo April 2011

Dev utils by Dickson S. Guedes June 2011




- By December 2010, the little fundraiser I set up had met its goal and it was time to get to work
- The site, pgxn.org, launched in April of 2011 with the first few extensions and a public REST API
- Meanwhile, Daniele Varrazzo wrote the command-line client in Python.
- And Dickson Guedes released a suite of Ruby dev utils in June 2011. The community involvement was so great!




Here's what pgxn.org looks like today. It currently hosts over 2300 releases of 404 extensions amongst 440 users. It's pretty cool, I think, you should check it out.


But alas, there has been little change since 2012. I would say, “In classic SDLC fashion, the PGXN POC shipped as an MVP and was neglected.”


- Meanwhile, things have not stood still the in the rest of the Postgres extensions ecosystem.

# CLOUDY DAYS

**Non-Core Extension Counts:**

**Azure: 25**

**GCP 29**

**AWS: 48**

**PGXN: 377**

**joelonsql/PostgreSQL-EXTENSIONs.md: 1,186**

- Chief among the changes has been the emergence of “Postgres as a service” providers. They curate extensions for their users. As of last spring…
- Azure provides 25 non-core extensions
- GCP provides 29
- AWS 48
- PGXN meanwhile has 377 distributions with 404 extensions
- But as I said, that’s a fraction of those available. Joel on SQL has inventoried almost 1200 publicly-available extensions


- This leads me to wonder: why has uptake been so…modest?

# LOST OPPORTUNITIES

**POSTGRES EXTENSIONS:**

**Difficult to find and discover**

**Under-documented and difficult to understand**

**Maturity difficult to gauge**

**Hard to configure and install**

**No comprehensive binary packaging**

**Centralized source distribution insufficient**

- Despite the best of intentions, some opportunities were never met.
- Postgres extensions remain quite difficult to find and discover, since they’re scattered to the internet winds
- Most are under-documented and difficult to understand
- When you do find them, it can be tricky to gauge the maturity, reliability, and stability of an extension
- Furthermore, they’re difficult to configure and install. Most users just want the add them to their dev and production clusters, and not have to install a bunch tooling like compilers and devel packages
- This is because there is no comprehensive binary packaging system covering a wide variety of platforms, architectures, and Postgres versions. It’s catch-as-catch-can between the community Apt and Yum repositories and tools like Homebrew and StackBuilder.
- Clearly PGXN’s centralized distribution of source code is insufficient to meet the needs of people who just want to quickly find, install, and use extensions.


But as I said, things haven’t stood still. A number of new projects have emerged in the last couple years that attempt to fill some of the development and distribution gaps.

# TLE

TRUSTED LANGUAGE EXTENSIONS

Goal: Empower app developers

Easy install via SQL functions

Portable (no compilation)

Hooks into `CREATE EXTENSION`

Supports custom data types

Plans for FDWs, background workers

From AWS & Supabase

- One such development is Trusted Language Extensions, or TLEs
- The TLE project wants to empower application developers to build database functionality without having to do the full compile/install dance on their servers
- The Pg_TLE extension provides functions for easy installation
- And TLEs can be ported between systems and architectures, since they require no compilation
- Once an extension is installed, pg_tle has hooks into CREATE EXTENSION to allow seamless loading
- The system supports custom data types via its API, which exposes the underlying Postgres APIs without requiring C access
- This pattern will be used to add support for other hooks like foreign data wrappers and background workers
- Our friends at AWS and Supabase steer this project

# PGRX

**Native Rust extensions**

**Build shared object libraries**

**Full access to Postgres features**

**Developer-friendly tooling**

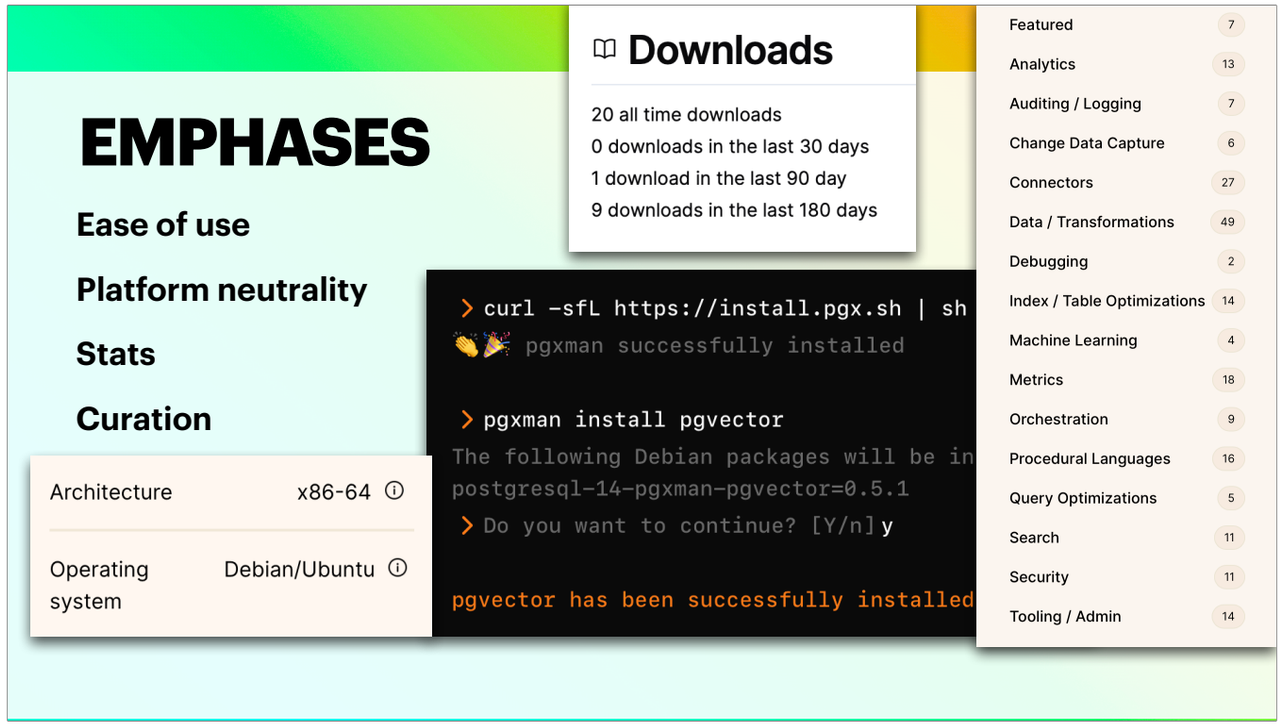
**Lots of community excitement**

**Under active development**

**From ZomboDB & PgCentral Foundation**

- Another recent entrant is pgrx,
- A framework for building extensions in Rust
- It’s like C in that it builds shared object libraries to be loaded into Postgres
- But it provides fairly transparent bridges from the Postgres C APIs to Rust
- pgrx also provides developer-friendly tooling to manage test clusters, generate SQL, and more
- There’s a ton of community excitement and a slew of new Rust extensions released every month
- The project itself is under active development
- Thanks to our friends at ZomboDB and the PG Central Foundation


- Meanwhile, a number of projects have cropped up to try to address discovery and installation issues, including DBDev, Trunk, and PGXMan.




- These binary registries have emphasized features that PGXN has not. These include
- Ease of use, as in this demonstration of installing PGXMan and a first extension in just two commands.
- Platform neutrality, as in Trunk showing the available platforms for an extension
- Statistics, such as DB Dev listing download number, which can be helpful for evaluating maturity and community acceptance
- And curation, as in Trunk’s use of categories, which have proven a popular vector for discovering extensions


All these developments have invigorated the broader extension ecosystem. But many of the promises we saw early on sadly have yet to be fulfilled.

# STATE OF AFFAIRS

**Despite early promise…**

**No single source**

**Many incomplete attempts**

**Poor discovery**

**Difficult installation**

**Insufficient docs**

**Low adoption**

**But…**

To whit

- There is no single source for a complete list of all extensions.
    - Every attempt and doing so remains incomplete
- This makes discovery difficult, as one has to know to search PGXN, GitHub, GitLab, Bitbucket, the Postgres Wiki, various cloud providers, specific web sites like postgis.org, and more
- Installation is difficult outside OS binary packages systems (if you can find a list of them). No one wants to compile extensions on their production database server.
- Most extensions provide very little documentation, usually just a README, and docs vary tremendously in quality and format
- All this has led to pretty low adoption rates. Few databases use extensions at all, and the number using more than one is vanishingly small
- Still, as we've seen…


“There sure has been a lot of excitement around extensions lately.”


Which brings us to the future. What new opportunities are there to improve the extensions ecosystem for developers and users?

# PGXN V2

**The New World Order**

**Started this year**

**Made possible by Tembo**

**Consider emerging patterns**

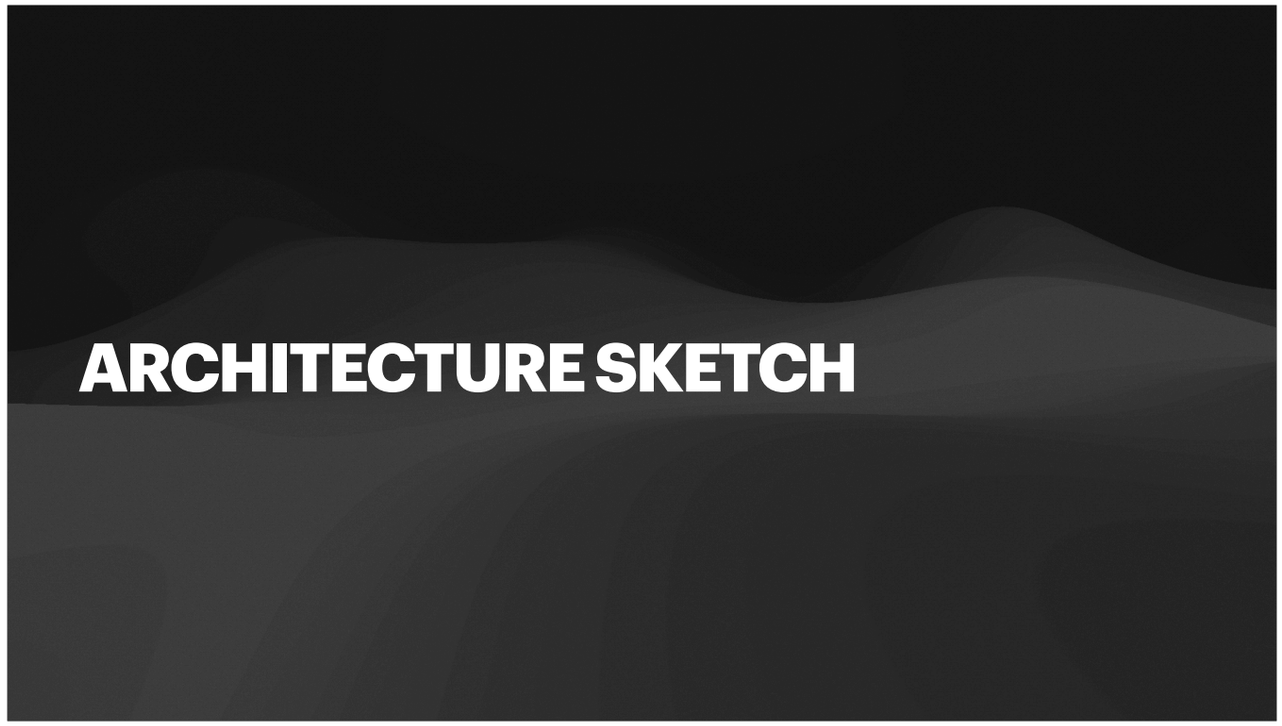
**Meditate on deficiencies**

**Engage deeply with the community**

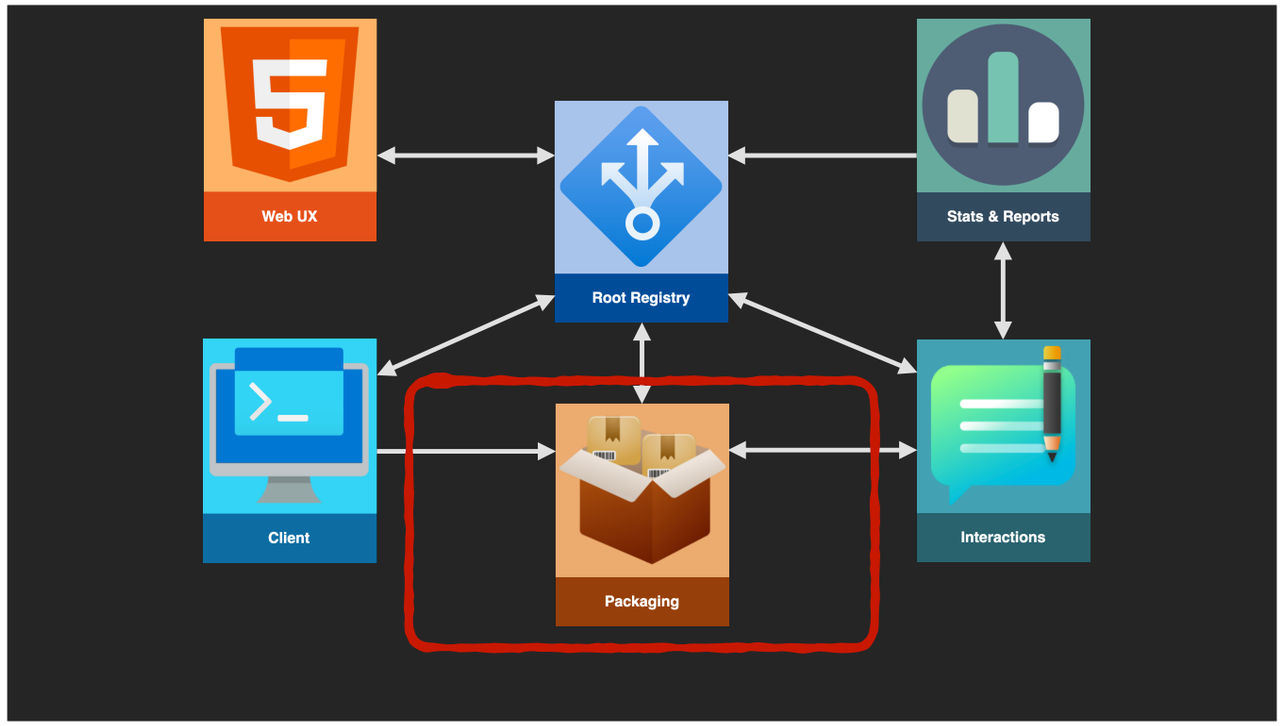
**Design and implement new architecture**

**Serve the community for next decade**

- To answer those questions, we've launched a project, code-named "PGXN v2"
- To find out and make it happen.
- This came about thanks to Tembo, the creators of Trunk, who hired me specifically to:
- Consider the emerging patterns, like binary registries and development frameworks
- Meditate on the deficiencies, including discovery and ease of installation
- To engage deeply with the broader community of Postgres developers, extension authors, and users
- In order to recognizer the jobs to be done and to design and implement a new architecture
- That will serve the community for the next decade or more




To that end I’ve sketched a high-level architecture to capture the categories of tools and services to fulfill jobs to be done.




- At root is, of course, the root source code registry
- This is complemented by a well-known web site where users can find extensions, read their docs, look at curation information, and download them
- Next is a command-line client which allows one to quickly find, download, compile, and install any of the indexed extensions.
- Brand new is the idea of Interactions, which stream releases and allow trusted third parties to submit curation data
- Stats and reports are the categories of services that provide that kind of data, like test matrixes and download numbers
- Last but not least we have binary packaging services that support a variety of OSes, architectures, and Postgres versions.
- In the last few months, we've been focused on the patterns to enable binary packaging

# METADATA & PACKAGING RFCS

https://github.com/pgxn/rfcs/pulls

Started developing RFCs

- Meta Spec v2
- JWS Release Signing
- Binary Distribution Format

Feedback appreciated!

Goals:

- META.json-driven packaging
- OCI distribution

- Over the spring & summer, we started developing and soliciting feedback on RFCs, including
  - An improved metadata specification for third-party dependency resolution, different types of extensions, and categorization & curation
  - A standard for signing PGXN releases, for improved validation and trust
  - A binary distribution format, inspired by Trunk and Python wheel, therefore called “Trunk”
- The immediate goals for these standards are
  - META.json-driven packaging, to enable automated packaging of any PGXN distribution
  - Distribution of those packages via standard Open Container Initiative registries, a.k.a. Docker registries.

# TRUNK OCI DEMO

To prove out the model, at least for Trunk packaging and OCI distribution, I worked up a POC.

# EOY 2024

# JOIN US

https://wiki.postgresql.org/wiki/PGXN_v2

- So that’s the vision. We’d love to have you join the effort. Feedback and ideas, in particular, are greatly appreciated.
- The home page for the project is in the Postgres Wiki, and links out to the project plan, architectural design, RFCs, blog posts, and more.

POSTGRESQL EXTENSION ECOSYSTEM SUMMIT EU

# THANK YOU

## STATE OF THE POSTGRES EXTENSION ECOSYSTEM

David E. Wheeler
PGXN, Tembo
2024-11-22