

# The Maze of Postgres Options

BRUCE MOMJIAN



This presentation explains the many decisions needed to deploy Postgres successfully.

*<https://momjian.us/presentations>*



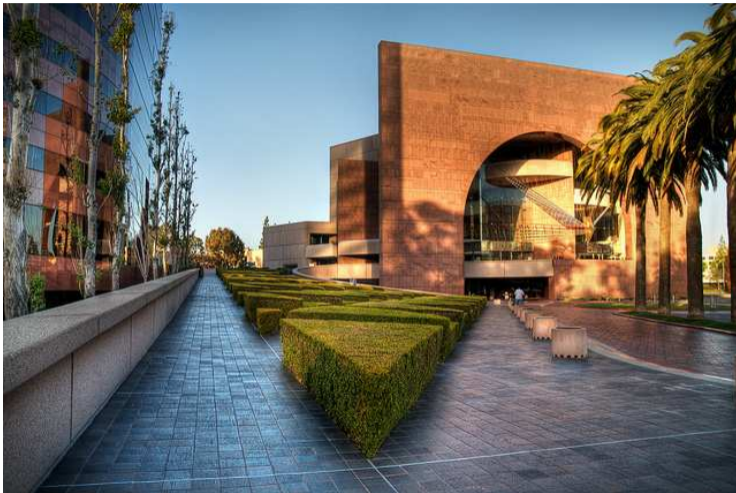
*Creative Commons Attribution License*

*Last updated: June 2024*

# Outline

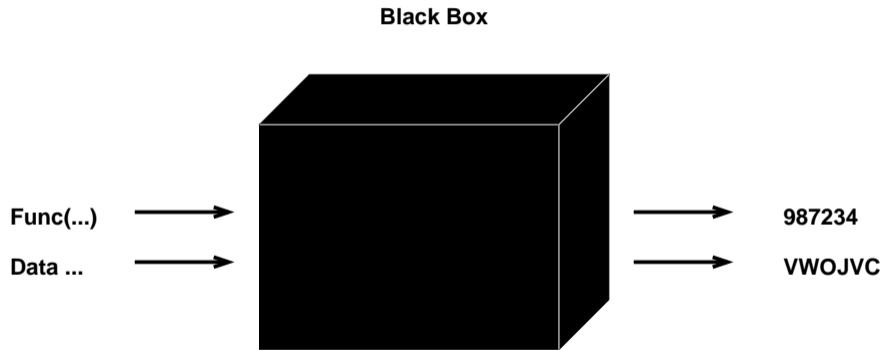
1. Open source's many options
2. Database server
3. Extensions
4. Deployment utilities
5. Monitoring options
6. Do you need a vendor?

# 1. Open Source's Many Options




[https://www.flickr.com/photos/zoso\\_tc/](https://www.flickr.com/photos/zoso_tc/)

# Closed-Source Software



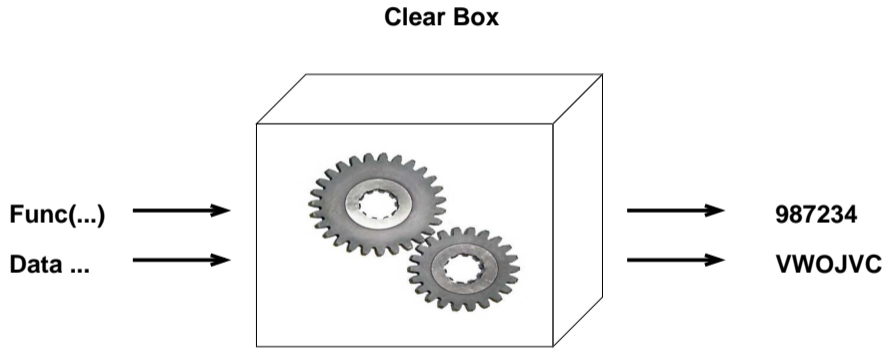
# Support of Close-Source Software

**Support**

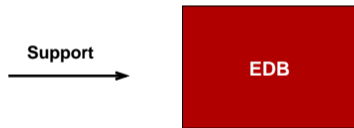


Database  
Company

# Open-Source Software



# Support of Open-Source Software



## 2. Database server

- Community server, [www.postgresql.org](http://www.postgresql.org)
- Company-enhanced (closed-source) servers
  - EnterpriseDB (EDB Postgres Advanced Server)
  - Postgres Professional (Postgres Pro Enterprise)
  - 2nd Quadrant (purchased by EDB)
  - FUJITSU (FUJITSU Enterprise Postgres)
  - Amazon Web Services (RDS Postgres)

These products closely track Postgres community versions.



## 3. Extensions

As an object-relational open-source database, extensions are a rich source of features in Postgres. These include:

- Extensions shipped with Postgres, e.g., *pgcrypto*, *postgres\_fdw*
- Extension repositories, e.g., PGXN
- Independent extensions, e.g., PostGIS, Citus extensions

# Server-Side Languages

- Included in the Postgres distribution
  - PL/Perl
  - PL/pgSQL
  - PL/Python
  - PL/Tcl
  - SPI
- Github
  - PL/Java
  - PL/PHP
  - PL/Ruby
  - PL/Scheme
  - PL/sh
- PGXN
  - PL/v8
- External
  - PL/R

## 4. Deployment Utilities

- Deployment utilities shipped with Postgres, e.g., *vacuumlo*
- Tool repositories, e.g., PGXN
- Company produced

# Backup Utilities

- Shipped with Postgres: *pg\_dump*, *pg\_restore*, *pg\_basebackup*
- pgBackRest
- Barman
- EDB Backup and Recovery Tool (BART)

# Failover Utilities

- Patroni
- repmgr
- EDB Postgres Failover Manager (EFM)

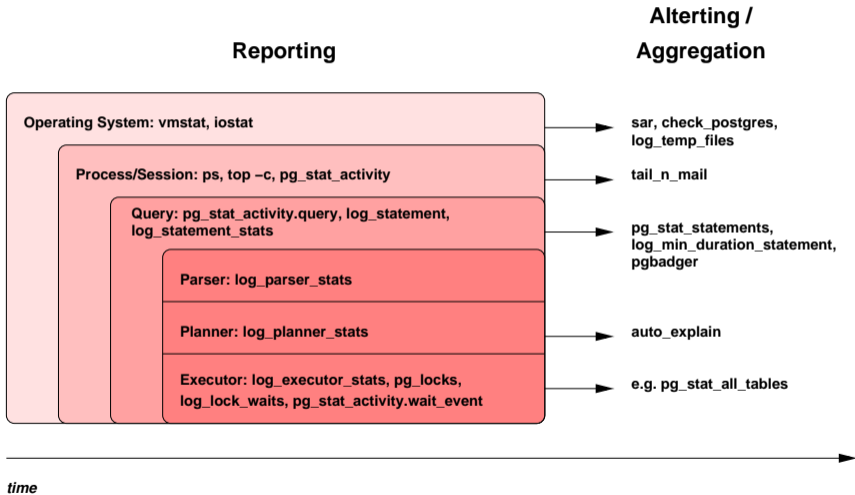
# Cloud/Container/VM Deployment

- Single-vendor cloud, e.g., AWS, Azure
- Multi-vendor cloud
- Docker deployment
- Kubernetes tools

## 5. Monitoring Options

- Shipped with Postgres, e.g., `pg_stat_activity`, `pg_locks`, statistic views, `pg_stat_statements`
- Independent, e.g., pgbadger, Nagios, Prometheus, Grafana
- Postgres Enterprise Manager (PEM)

# Monitoring Capabilities





## 6. Do You Need a Vendor?

- The database server and many extensions and tools are available for free
- Adding the missing parts is “a simple matter of programming”
- You can build your own Postgres open source support team

This is only possible because open source gives everyone the ability to become an expert.

## Becoming an Expert Has Costs

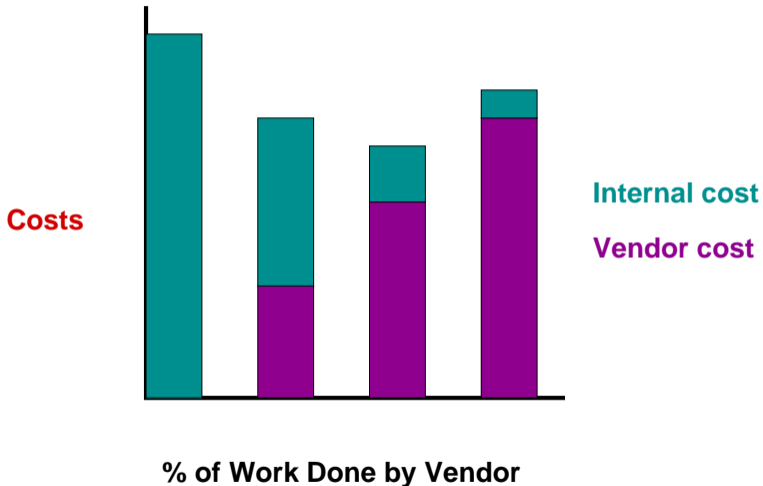
- Payroll, management
- Testing, risk
- Delayed deployment
- Increased dependency on staff skills
- Deployment rigidity, but easy customization

# Make or Buy Decision

Cook for yourself or go to a restaurant?

- Cook yourself
  - Ingredients
  - Cooking skills
  - Single dish
  - Less cost, more time
- Restaurant
  - Variety
  - Simplicity, flexibility
  - More cost, less time

# In-House/Vendor Trade Off



## Choosing a Postgres Vendor (or Restaurant)

- A broad range of vendor-cost options are available, more than in proprietary software
- Are vendor expenses less than the value of increased efficiency and reduced in-house costs?
  - Paid vendor \$400k but increased efficiency and reduced in-house costs by \$700k
- More expensive vendors can sometimes increase efficiency and reduce in-house costs more than inexpensive vendors

# Postgres Vendor Specialties

- Support
- Consulting
- Open-source products
- Closed-source products
- Cloud hosting
- Training
- Migration assistance
- System health check

# Conclusion



<https://momjian.us/presentations>

<https://www.flickr.com/photos/judy-van-der-velden/>