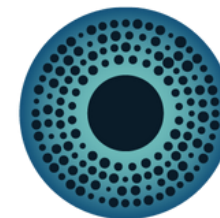




PostgreSQL

Migrating to PostgreSQL

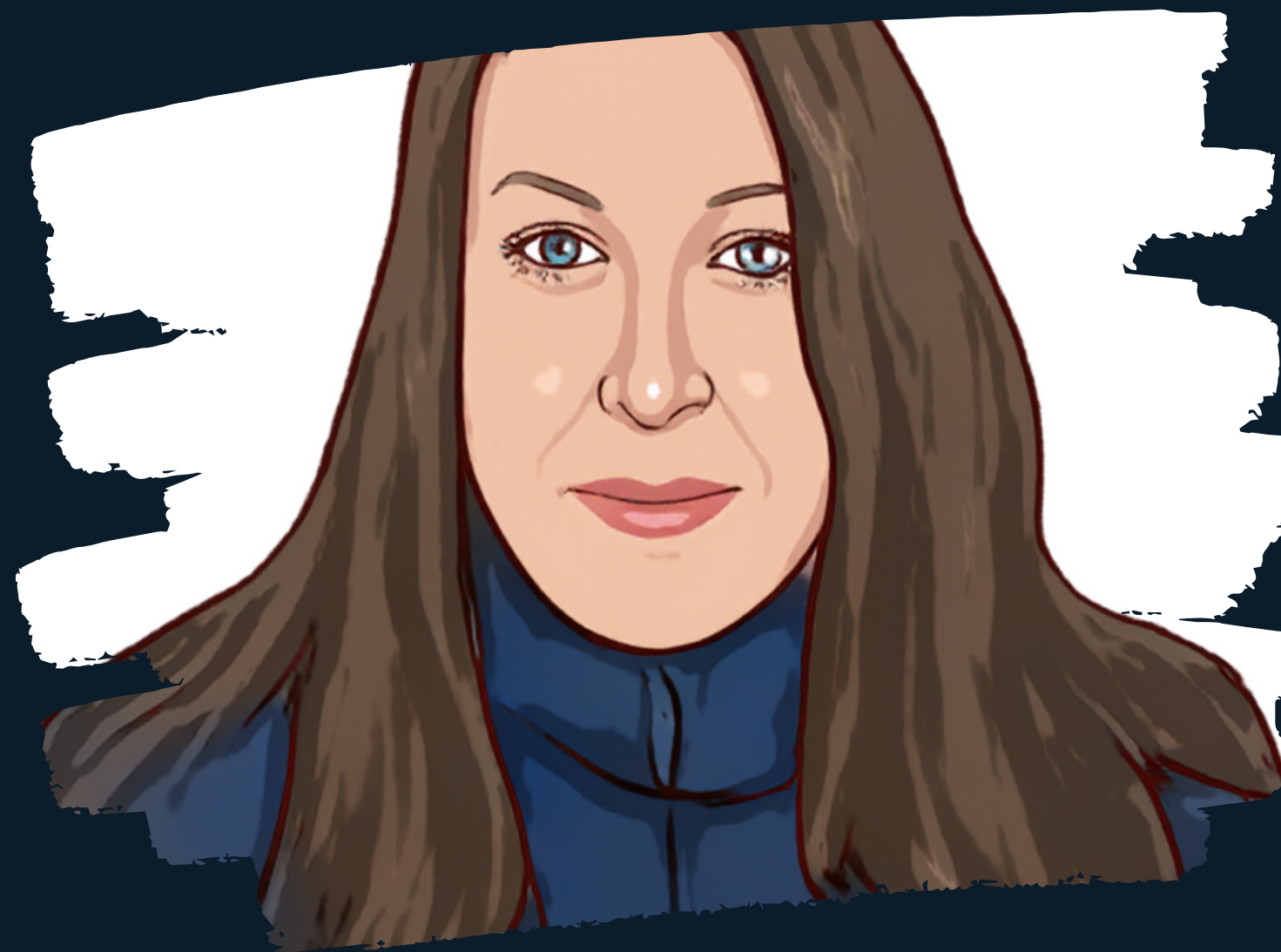


**Sharper
Informatics
Solutions**

@SharperInfo  
#SharperInfo #SCaLE19x

Hello!

At Sharper Informatics Solutions, we believe that smart informatics can drive good decision-making and open doors to new opportunities. We specialize in creating custom data-driven systems for clients in energy efficiency, data science and engineering.



Jennifer Scheuerell

CTO and Database Software Chameleon
at Sharper Informatics Solutions

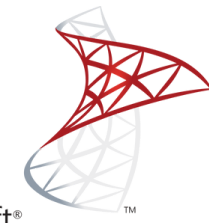
Comparing RDBMS

Key factors

- Database administration overhead
- Development expense
- Software cost
- Performance
- Integration
- Tools



PostgreSQL



Microsoft®
SQL Server®

ORACLE



MySQL®



SQLite

Why choose PostgreSQL



Security

Performance



Reliability

Integration Tools

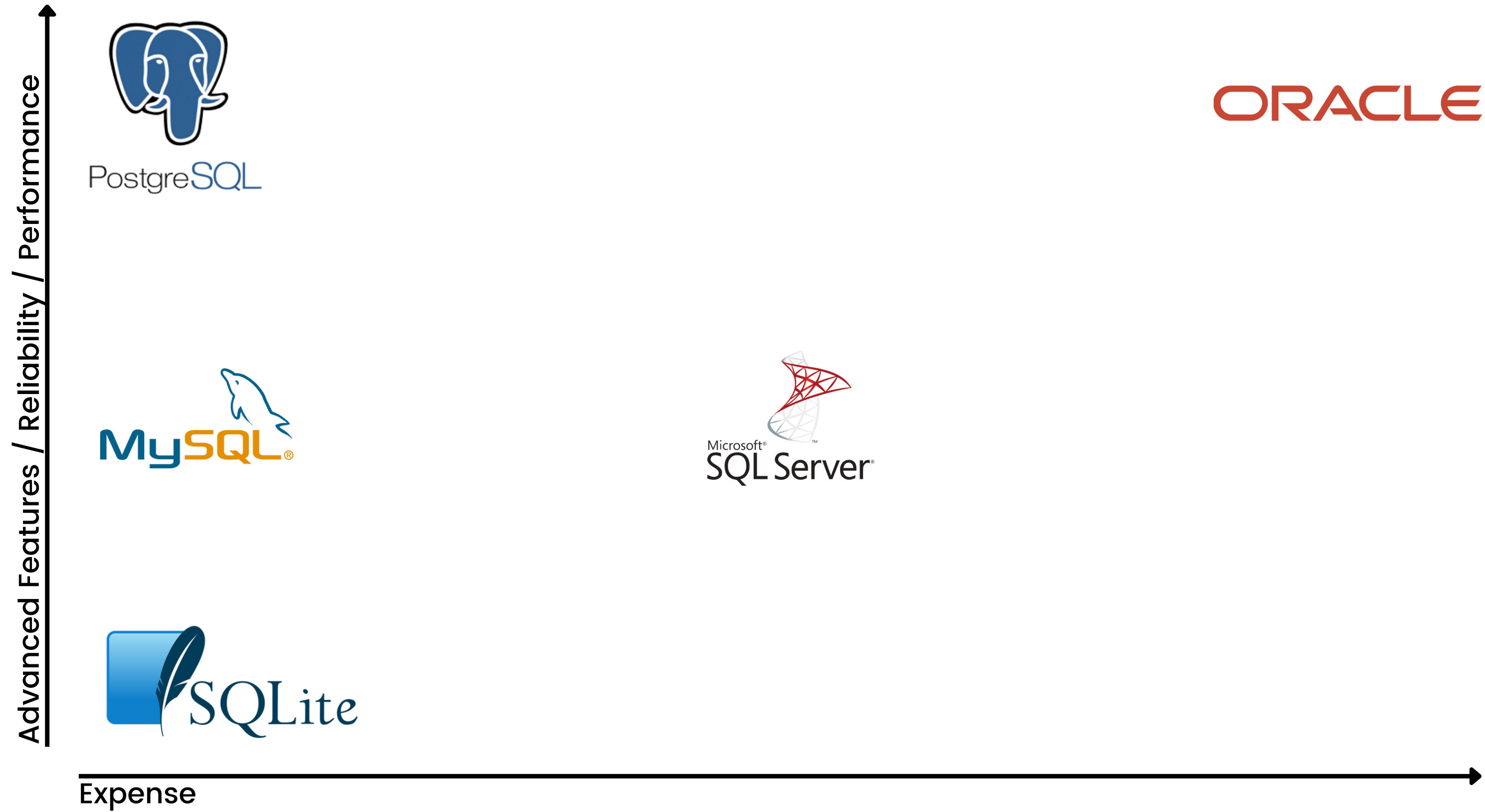


Automation

Cost



Migrating to PostgreSQL



Increasingly popular

PostgreSQL is the DBMS of the Year 2020

by [Paul Andlinger](#), [Matthias Gelbmann](#), 4 January 2021

Tags: [DBMS of the year](#), [Microsoft Azure SQL Database](#), [MongoDB](#), [PostgreSQL](#)

[PostgreSQL](#) is the database management system that gained more popularity in our [DB-Engines Ranking](#) within the last year than any of the other 360 monitored systems.

*We thus declare **PostgreSQL** as the **DBMS of the Year 2020**.*

For determining the DBMS of the year, we subtracted the popularity scores of January 2020 from the latest scores of January 2021. We use the difference of these numbers, rather than a percentage, because that would favor systems with a tiny popularity at the beginning of the year. The result is a list of DBMSs sorted by how much they managed to increase their popularity in 2020, or in other words, how many additional people started to communicate about it in one of the ways we measure in our [methodology](#), for example job offers, professional profile entries and citations on the web.

DBMS of the Year: PostgreSQL



[PostgreSQL](#) already won our DBMS of the Year award in 2017 and 2018, and now becomes the first system to win this title three times.

Continuous Migration



PGLOADER

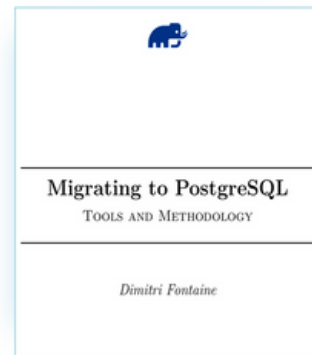
BLOG ABOUT LICENSING ROADMAP SERVICES WHITE PAPER

CONTINUOUS MIGRATION

Dec 21, 2017 · Dimitri Fontaine · 8 minute read

After having been involved in many migration projects over the last 10 years, I decided to publish the following [White Paper](#) in order to share my learnings.

The paper is titled [Migrating to PostgreSQL, Tools and Methodology](#) and details the **Continuous Migration** approach. It describes how to migrate from another relational database server technology to PostgreSQL. The reasons to do so are many, and first among them is often the licensing model.



From MySQL to PostgreSQL over the Week-End!

On February the 18th, 2015 I received a pretty interesting mention on Twitter:



The migration method proposed here is called *Continuous Migration*. Continuous Migration makes it easy to make incremental progress over a period of time, and also to pause and resume the migration work later on, should you need to do that. The method is pretty simple — just follow those steps:

1. Setup your target PostgreSQL architecture
2. Fork a *Continuous Integration* environment that uses PostgreSQL
3. Migrate the data over and over again every night, from your current production RDBMS
4. As soon as the CI is all green using PostgreSQL, schedule the D-day
5. Migrate without any surprises... and enjoy!

This method makes it possible to break down a huge migration effort into smaller chunks, and also to pause and resume the project if need be. It also ensures that your migration process is well understood and handled by your team, drastically limiting the number of surprises you may otherwise encounter on migration D-day.

Migrating to PostgreSQL

Case Studies

CEDARS

Advice Letters

Energy Data Web

01

Data problem

What motivated the migration? How was it approached?

02

Migration

Handy tools and techniques

03

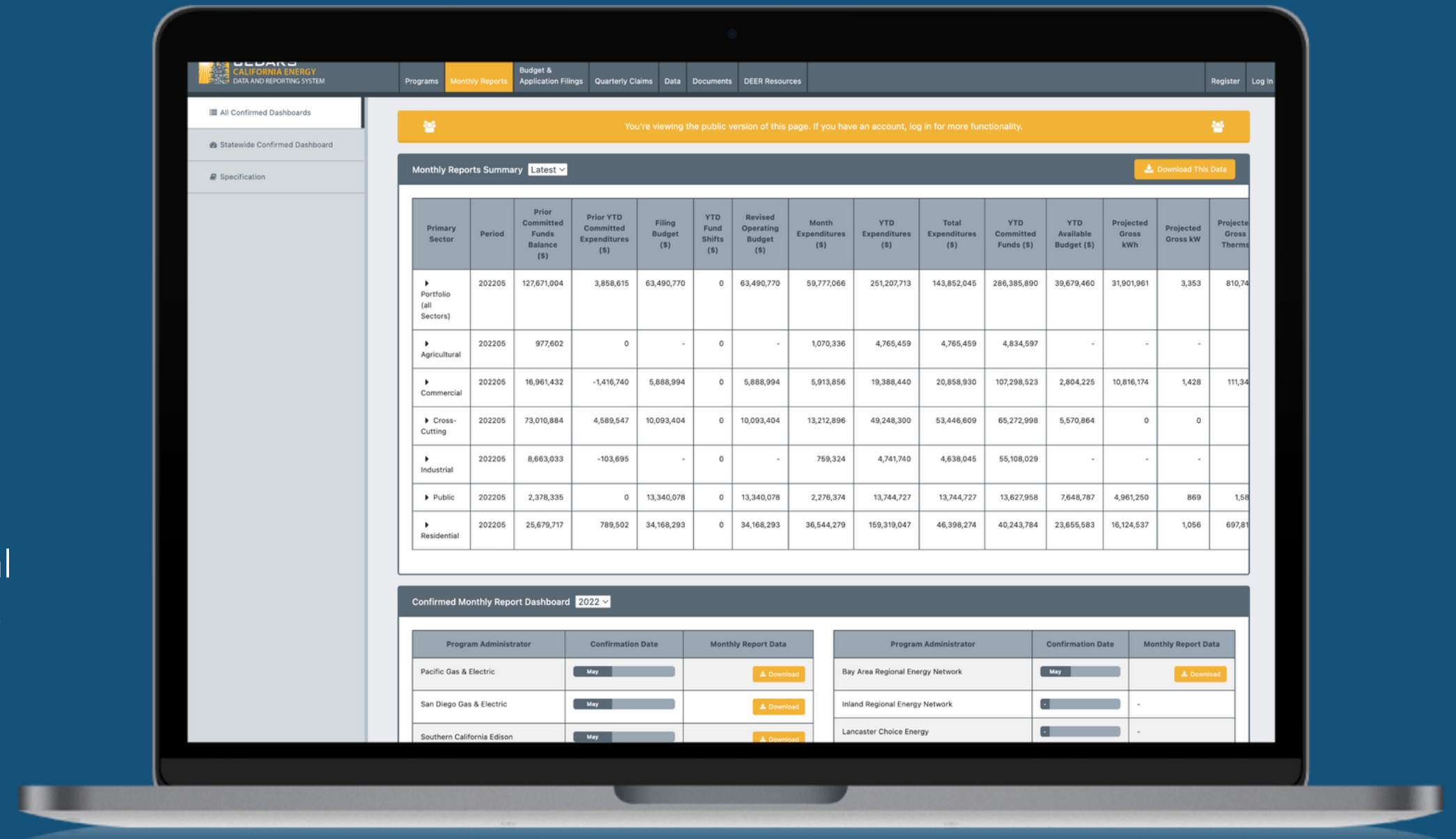
Lessons

What we've learned

Migrating to PostgreSQL

CEDARS

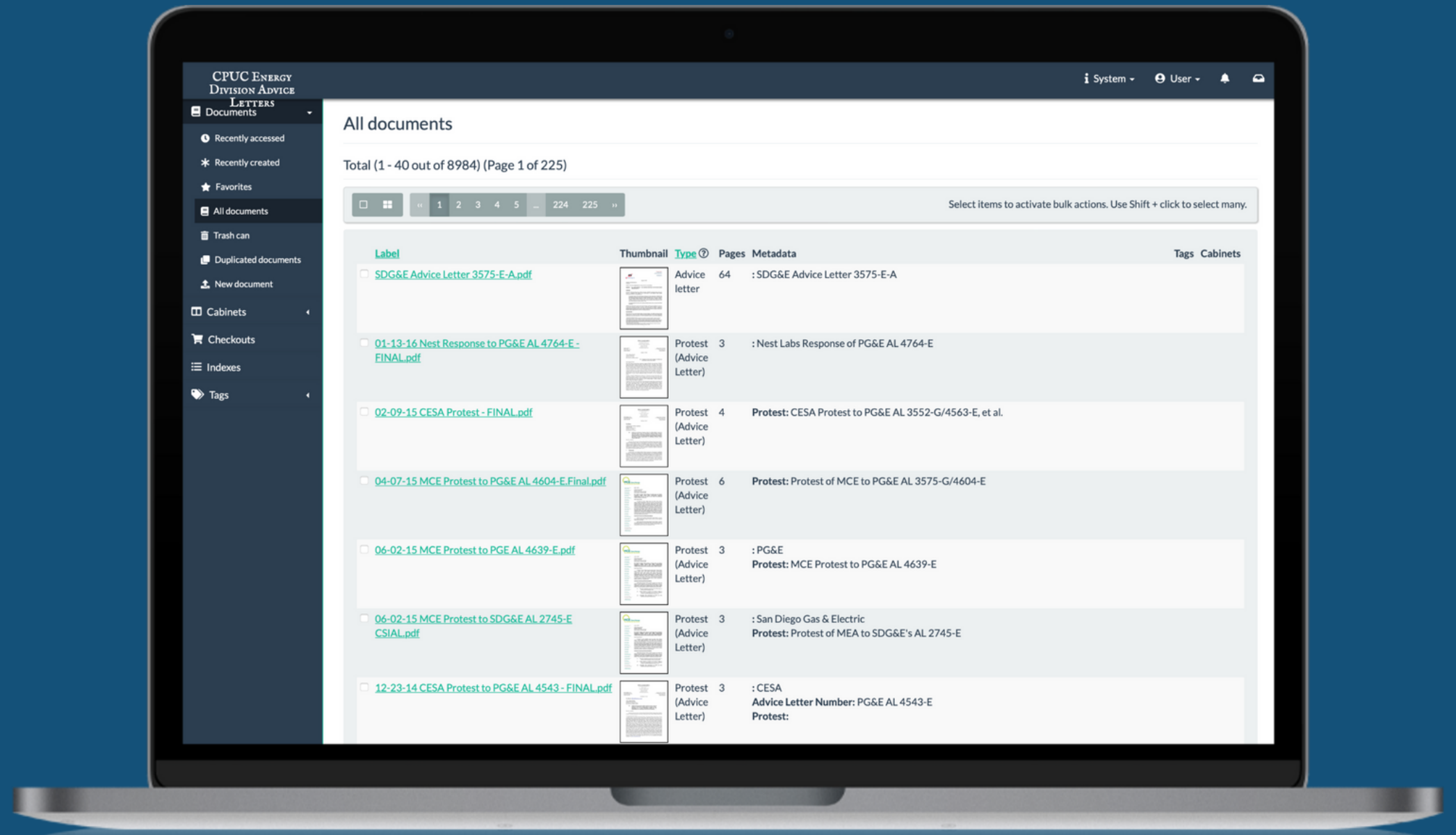
CEDARS securely manages California Energy Efficiency Program data reported to the Commission by Investor Owned Utilities, Regional Energy Networks (RENs), and certain Community Choice Aggregators (CCAs.)



Migrating to PostgreSQL

Advice Letters

CPUC Advice Letters database for Energy Utilities



Migrating to PostgreSQL

Energy Data Web

CPUC public-facing document and project sites

ENERGY PROJECT STATUS REPORTING SYSTEM

Add Project Support User Guide Login Register Links

This page provides quick access to basic information of currently planned evaluation activities conducted by the CPUC and utilities and further described in the joint evaluation plan.

Activity Report +

Project Status

Filter Projects By: Case Sensitive

Filter Specific Columns [reset to defaults](#)

Funding Cycle Project Status

Study Lead Study Type

CALIFORNIA ENERGY EFFICIENCY ENERGY CONTRACTS

Links Support User Guide Log

Evaluation Studies Public Document Search

Welcome to the Energy Division's Public Document Area. At this location you can find:

- The current master evaluation plan
- Specific research plans
- Draft reports, and
- Provide comments on active documents

These materials are made available to support California's Evaluation, Measurement, and Verification (EM&V) work. You can also subscribe to get regular updates. Work products developed by CPUC and/or Program Administrators are available on this site.

Search

Search Documents Title Search

[Advanced Search](#)

Search Results

Search Results List:

	Comments End	Published
July 2022 Evaluation, Measurement, and Verification (EM&V) Quarterly Stakeholder Webinar		7/26/22

The Energy Division hosted an Evaluation, Measurement, and Verification (EM&V) Quarterly Stakeholder webinar on Thursday, July 21st from 9:30 AM to 3:20 PM. The meeting materials are now available.

Generic Migration Steps

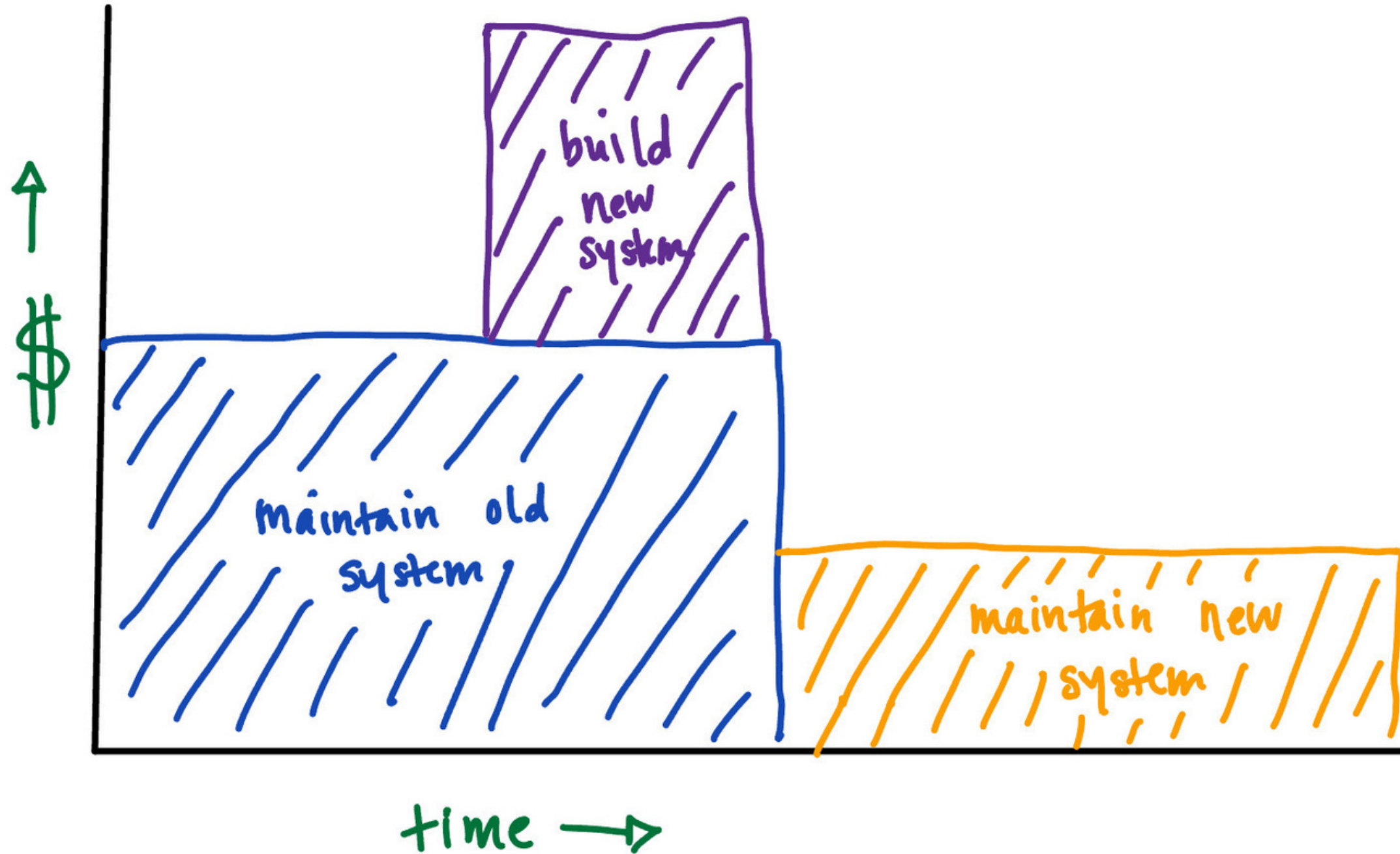
1. Agree to undertake the migration
2. Set aside time and budget
3. Plan new servers
4. Build out new servers
5. Set up and start continuous migration
6. Set up backups, automate recovery, test
7. Schedule cut over
8. Run final migration
9. Retire old servers

Migrating to PostgreSQL

Deciding to migrate

Do you wait for "the incident" or migrate before it happens?

Dedicating Resources



Generic Migration Steps

1. Agree to undertake the migration
2. Set aside time and budget
3. Plan new servers
4. Build out new servers
5. Set up and run continuous migration
6. Set up backups, script recovery, test
7. Schedule cut over
8. Run final migration
9. Retire old servers



Server expense

Is Oracle DB expensive? ^

With one of its products licensed at \$300,000 per Oracle Processor. Oracle Business Intelligence Suite Foundation, licensing a relatively small 4-processor quad-core server would cost \$2.4 million (not including support and maintenance), making it **one of the world's most expensive software products to license.** Jan 31, 2017

<https://www.linkedin.com/pulse/oracle-most-expensive...>

[Oracle, the most expensive software on the planet? - LinkedIn](#)



Server expense

We run three PostgreSQL servers on Linux for less cost than one MS SQL Server database on Windows using the same hardware.



Reliability

We saw significant site performance degradation over time on SQLite.

The Windows MS SQL Server crashed and took five days to restore.

Integration Tools

Foreign data wrapper

```
CREATE EXTENSION IF NOT EXISTS postgres_fdw;

CREATE SERVER etrm FOREIGN DATA WRAPPER postgres_fdw OPTIONS (
    port '5432',
    servername 'publicdb.caetrm.com'
);

CREATE USER MAPPING FOR postgres
    SERVER etrm
    OPTIONS (username 'readonly', password '3PIAFrEdkV');

CREATE FOREIGN TABLE etrm_permutations (
    MeasDetailID text)
    SERVER etrm
    OPTIONS(query
        'SELECT distinct "MeasDetailID" as measdetailid
        FROM public.permutations');

create materialized view main_measdetailid as
    SELECT measdetailid
    FROM etrm_permutations;
```





Performance

Overall better performance

Materialized views

Powerful tool

<https://pgloader.io/>
<https://github.com/dimitri/pgloader>

pgloader loads data into PostgreSQL and allows you to implement Continuous Migration from your current database to PostgreSQL.

```
$ apt-get install pgloader
```



PGLOADER
BLOG ABOUT CONTRIBUTE ROADMAP WHITE PAPER

pgloader loads data into PostgreSQL and allows you to implement [Continuous Migration](#) from your current database to PostgreSQL. Read the [White Paper](#) to learn how to limit risks and control your budget, and start your PostgreSQL migration today!

Introduction



pgLoader has two modes of operation. It can either load data from files, such as CSV or Fixed-File Format; or migrate a whole database to PostgreSQL.

pgLoader supports several RDBMS solutions as a migration source, and fetches information from the catalog tables over a connection to then create an equivalent schema in PostgreSQL. This means that you can [migrate to PostgreSQL in a single command-line!](#)

Supported operations include:

- Migrate from MySQL to PostgreSQL
- Migrate from SQLite to PostgreSQL
- Migrate from MS SQL Server® to PostgreSQL

You can also *migrate from database files* in the DBF and IXF formats, where pgLoader can inspect the target table format for you automatically in the file headers.

Why choose PostgreSQL



Security

Performance



Reliability

Integration Tools

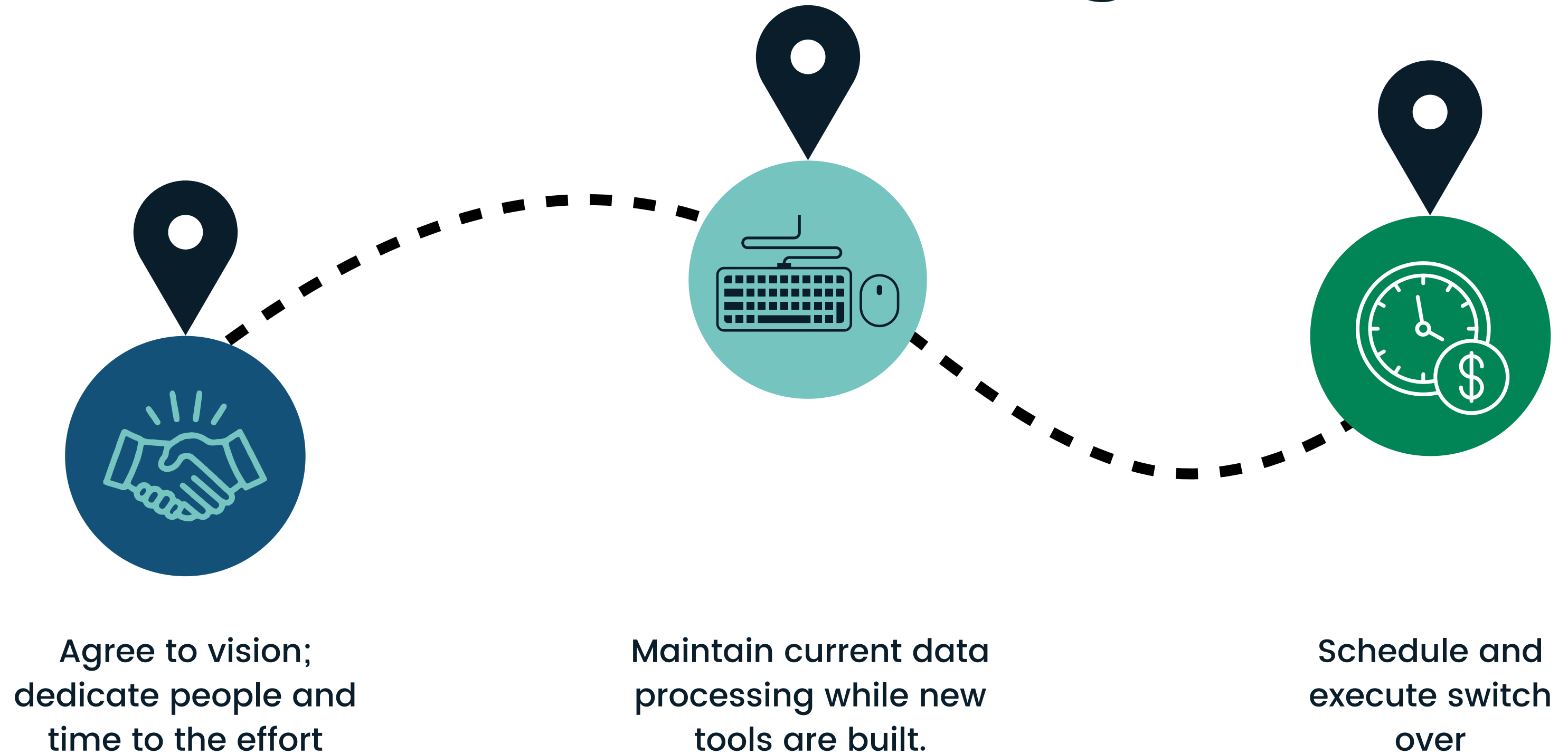


Automation

Cost



Hurdles for management



Conclusion

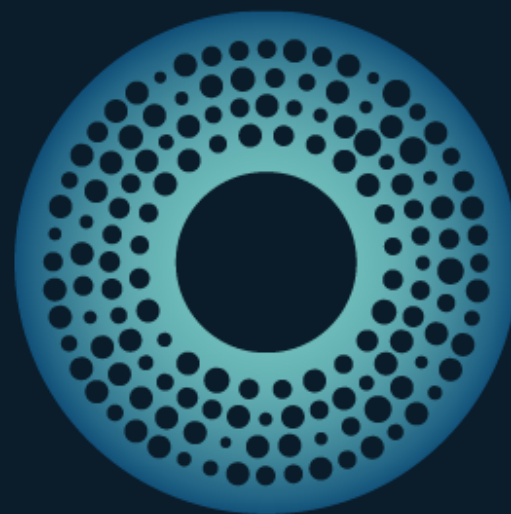
PostgreSQL for the win

- Happy clients
- Lighter database administration overhead
- Increased reliability
- Better performance
- Powerful integrations
- Much lower cost



Migrating to PostgreSQL

THANK YOU!



**Sharper
Informatics
Solutions**

sharperinfo.com

humans@sharperinfo.com



@SharperInfo

#SharperInfo #SCaLE19x