GSoC 2024

Enhancing Data Insights for Postgres with Remote Sinks

Presented By
Akshat Jaimini & Pavlo Golub

$ *whoami* -- mentor

- Senior Developer at Cybertec
- Co-Founder at PostgreSQL Ukraine
- Mentor and Postgres Org Admin at GSoC
- pgwatch maintainer
- Get in touch!
  - pashagolub.github.io
  - cybertec-postgresql.com/en/blog/

- 2024 marks the 20th anniversary of Google Summer of Code
- PostgreSQL has been a proud participant for 17 years since 2008
- A long-standing member of the GSoC family!

Mentorship and Admin Roles

Mentorship is crucial for:

- Spreading knowledge
- Bringing in new contributors
- Growing PostgreSQL and related projects

I've had the privilege of serving as an admin and mentor

Community Impact

PostgreSQL org is an "Umbrella" project:

- Supports the entire "PostgreSQL Family"
- pgmoneta, pgagroal, PgJDBC, WAL-G, pgBackRest, pgwatch, pgpool, pgAdmin, and more...
- GSoC participation strengthens the broader PostgreSQL ecosystem

PostgreSQL 2024 Projects

- 5 projects in 2024:
  - pgmoneta: WAL infrastructure by Shahryar Soltanpour
  - PostgreSQL JDBC Struct/Array Support by Arjan Marku
  - RPC Sinks for PgWatch3 by Akshat Jaimini
  - pgmoneta: Extended functionality by Chao Gu
  - pgagroal: Replace the I/O Layer by Henrique A. de Carvalho

2023 GSoC Stats

- Global impact in 2023:
  - 966 contributors from 65 countries
  - 1,954 mentors with active projects from 77 countries
  - 168 open source organizations
  - 93.48% overall success rate

Looking Ahead to GSoC 2025

- We need:
  - Mentors!
  - Project ideas!
  - Contributors!
- How can you help?
  - Mentor, suggest ideas, spread the word
- Join us in fostering the collaborative PostgreSQL community!

Today’s Agenda

1. What is PgWatch?
2. Understanding Sinks
3. Why ‘Remote’ Sinks?
4. Remote Sinks Architecture

5. Developing Custom Sinks

6. Setting up Remote Sinks with PgWatch v3

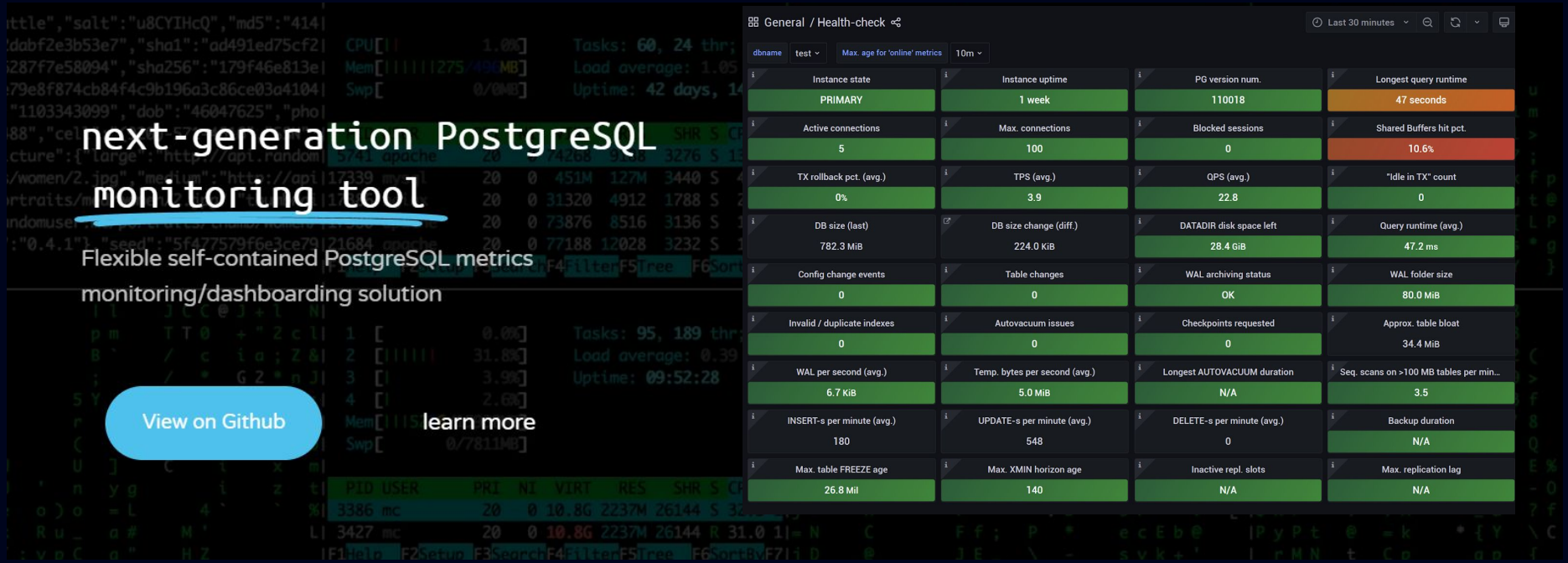
7. *Live Demo!*

Today's Agenda

$ *whoami* -- contributor

- Computer Engineering Undergrad, from TIET, India - graduating in 2025
- Joined the PostgreSQL community through GSoC@2023
- Maintaining Pgweb Testing Harness
- Contributing to the wider PG ecosystem

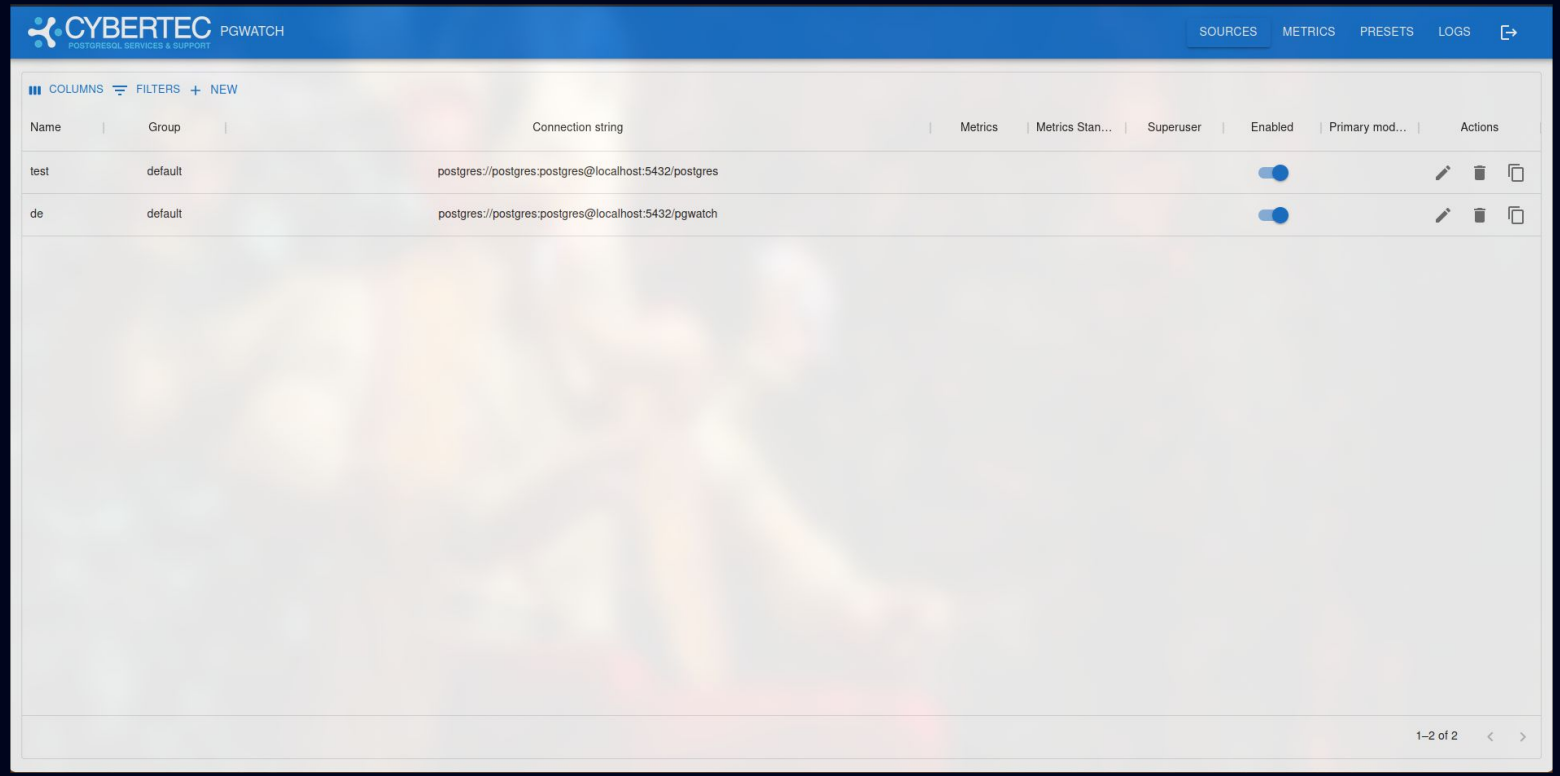
What is PgWatch?



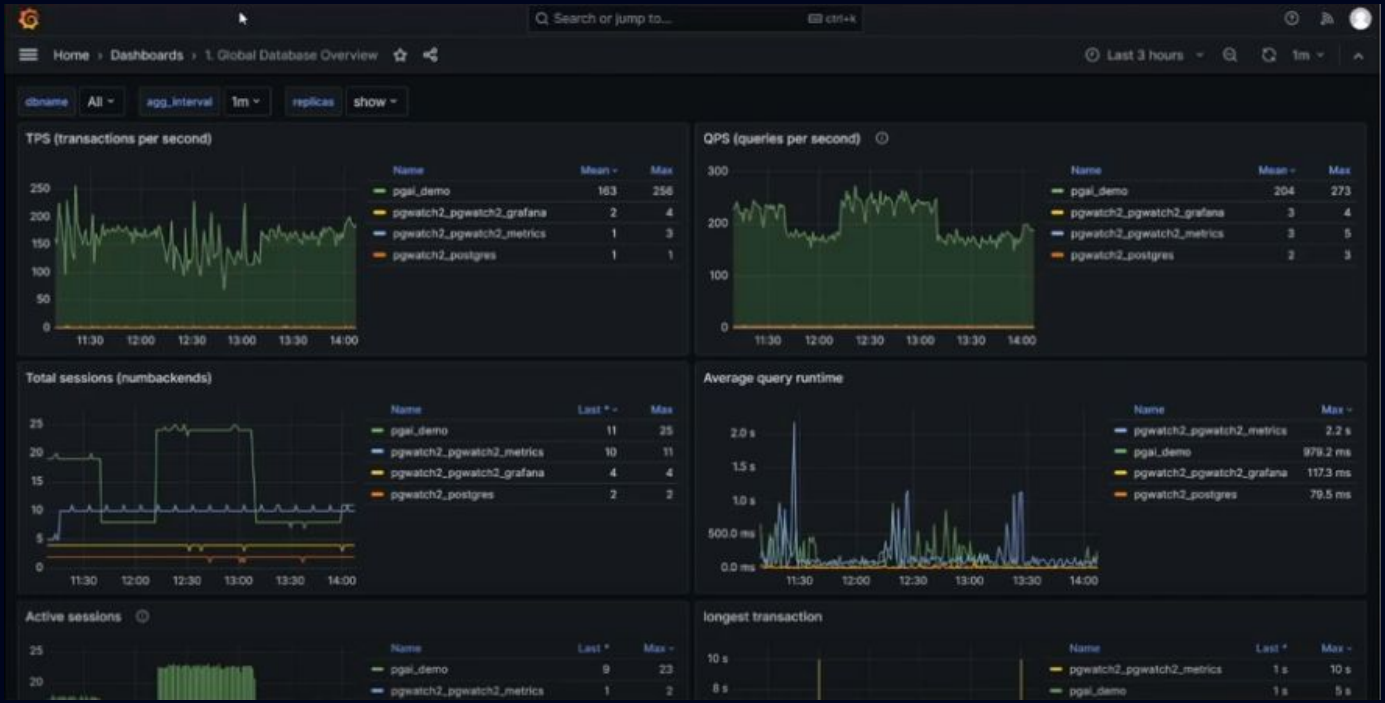
Some Cool Features in PgWatch v3

- Easy setup using Docker
- Support for existing metrics + Custom Metric Definitions in SQL
- Very low resource consumption
- Grafana Dashboards for monitoring UI
- Support for Alerting via Grafana and pg_timetable scheduler
- Supports ability to export measurements

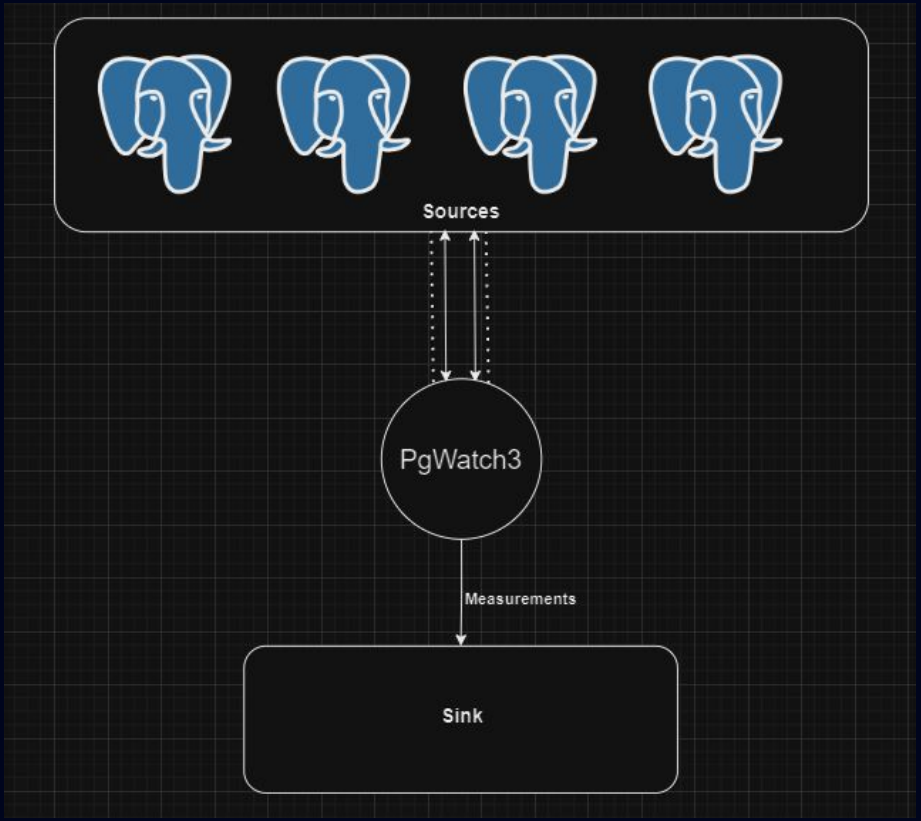
```
$ docker compose -f
./docker/docker-compose.yml up --detach
```



PgWatch3 Management UI

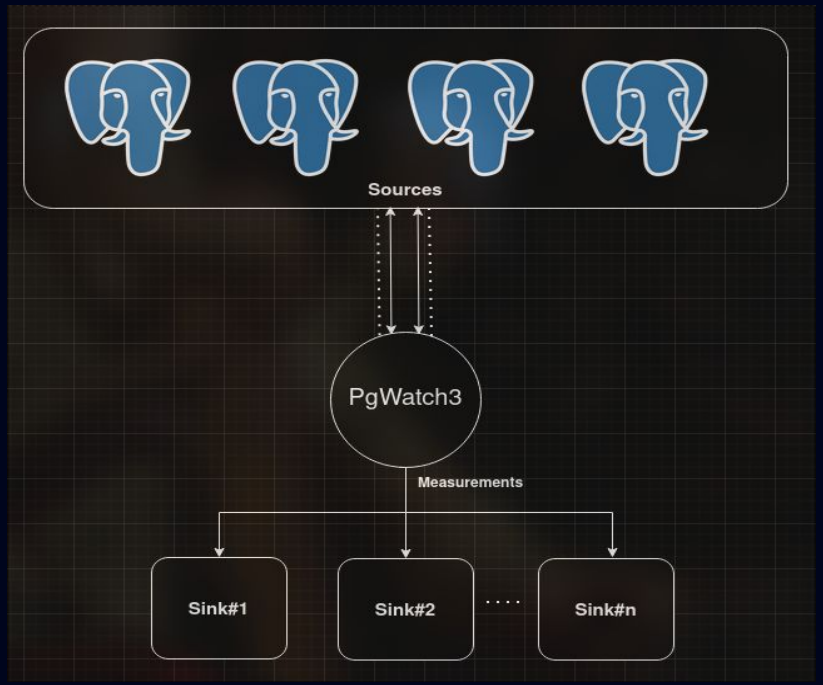


Grafana Dashboard for PgWatch3



3 Pillars Of PgWatch:

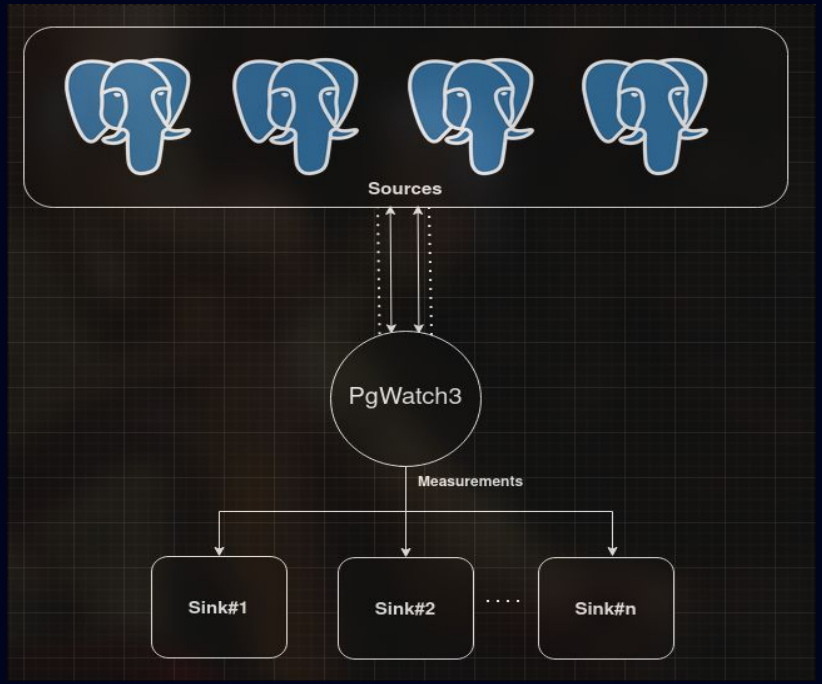
- *Sources:* Postgres Databases to monitor
- *Metrics:* Defined parameters to extract from DB
- *Sinks:* To store measurements taken by pgwatch



Change from V2

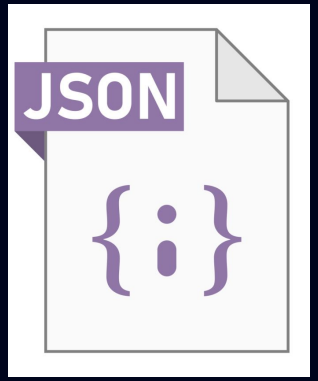
- *Pillars Decoupled in implementation*
- *Parallel Sinks*

Understanding Sinks



- *Sinks:* To store measurements taken by pgwatch
- *Parallel Config:* You can even use multiple sinks in parallel with pgwatch

postgres://....

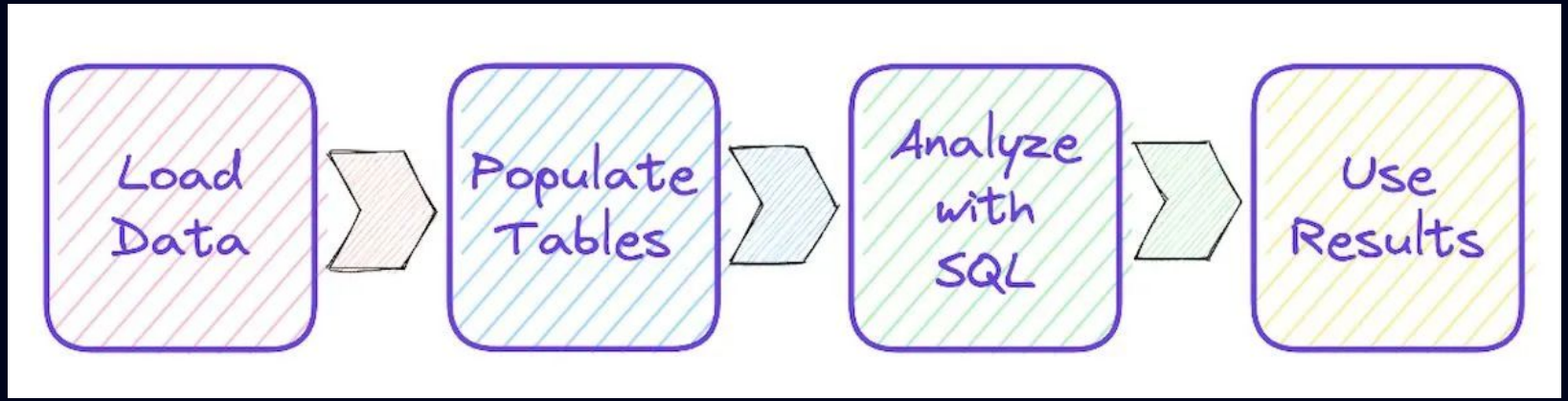


json://....

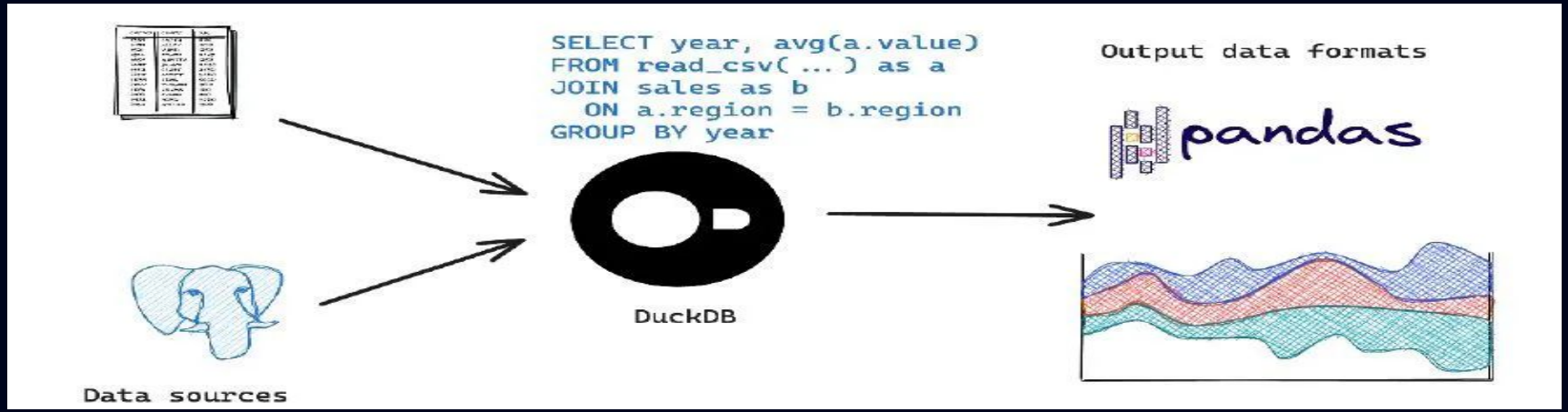
prometheus://....

Supported Sink Formats in pgwatch

What if I want to use something else as my sink?

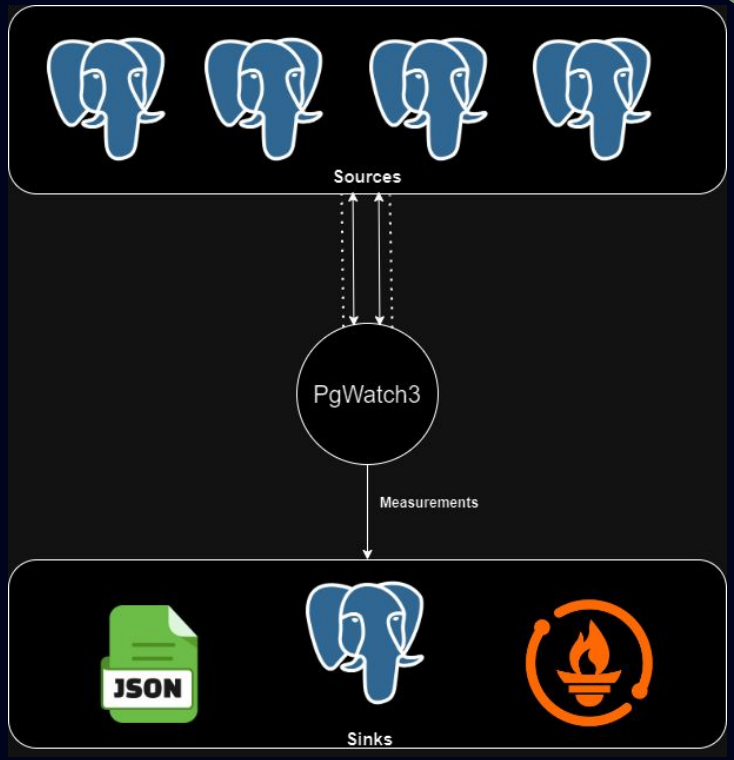


What if I want to do something else with my measurements?



Desired Solution...

- *Solution should be easy to use*
- *It should be easy to maintain*
- *It should reduce overhead for users*

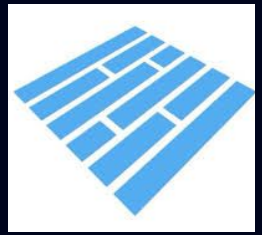


Limited Options in Sinks !!

Solution #1

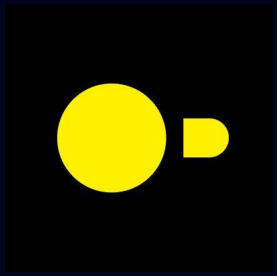
Add support for every Possible storage out there!!

(What could possibly go wrong.....?)



JUST TOO MANY TOOLS & STORAGES !!!!!

• • • •



Well…..
It’s a little hard to maintain

Solution #2

Maybe a contrib repo...?

(Just Like Open Telemetry!)

Again Same Problems....

Too many implementations to Maintain

&

Everyone has different motives behind a single implementation

Original Impl - One parquet file for All Databases

Sink.parquet

Db Name	Metric Name	Measurement	Timestamp
DB#1	db_size	{.....}	1728057357
DB#1	locks	{.....}	1728057358
DB#1	db_stats	{.....}	1728057359
DB#2	db_size	{.....}	1728057360
DB#2	locks	{.....}	1728057361
⋮	⋮	⋮	⋮
DB#N	db_size	{.....}	1728057357

What if someone wants data for a single DB only?

One parquet file per database

User#1

User#2

DB1.parquet

Metric Name	Measurement	Timestamp
db_size	{.....}	1728057357
locks	{.....}	1728057358
db_stats	{.....}	1728057359

DB2.parquet

Metric Name	Measurement	Timestamp
db_size	{.....}	1728057360
locks	{.....}	1728057361

What if I want data for a single Metric only?

One parquet file per Metric

User#1

User#2

db_size.parquet

Db Name	Measurement	Timestamp
DB#1	{.....}	1728057357
DB#2	{.....}	1728057358
DB#3	{.....}	1728057359

locks.parquet

Metric Name	Measurement	Timestamp
DB#1	{.....}	1728057360
DB#2	{.....}	1728057361
DB#3	{.....}	1728057360

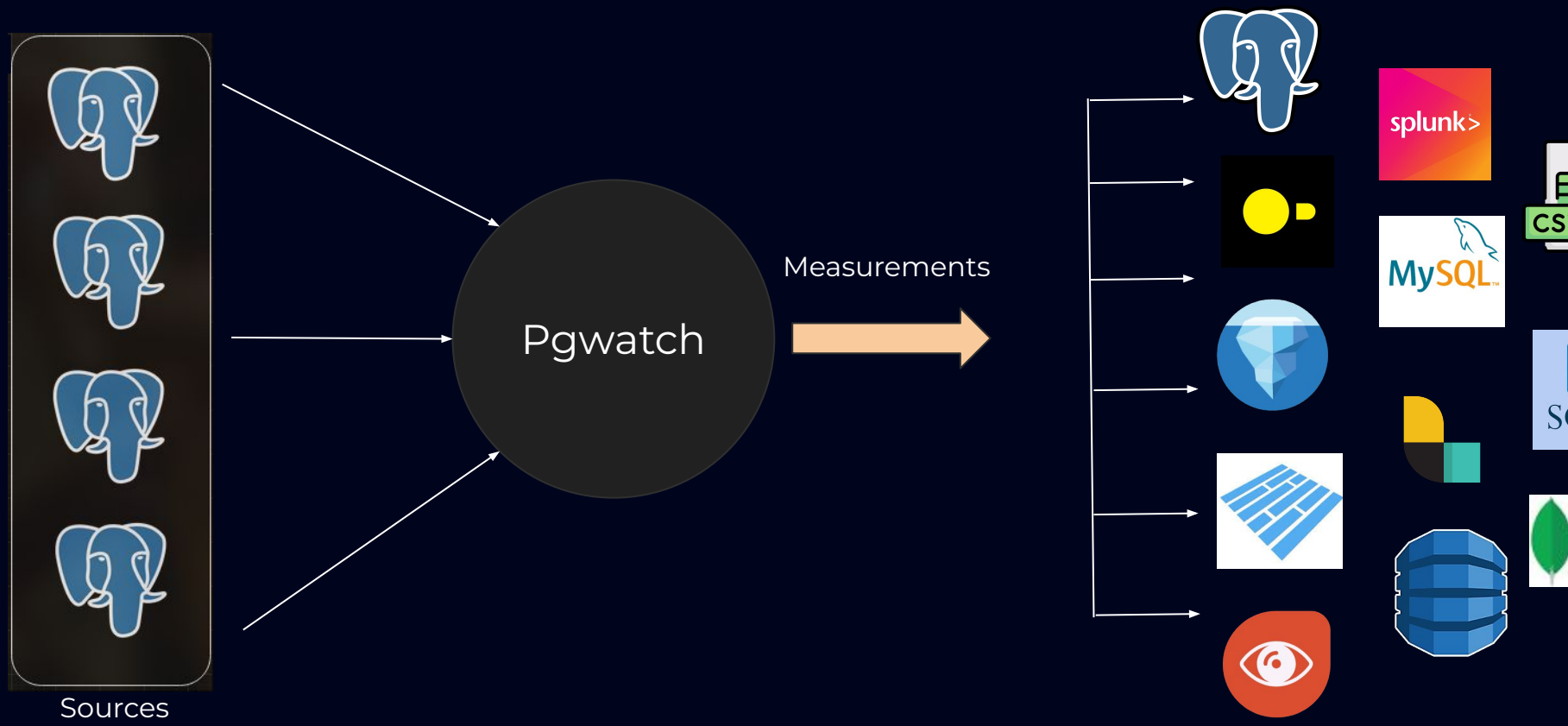
Final Solution

Let’s give our *users* the power to do that!!!

Abstractions for the win!

Introducing Remote Sinks

Remote Sinks Architecture



Remote Sinks Architecture

Sources

Pgwatch

Measurements

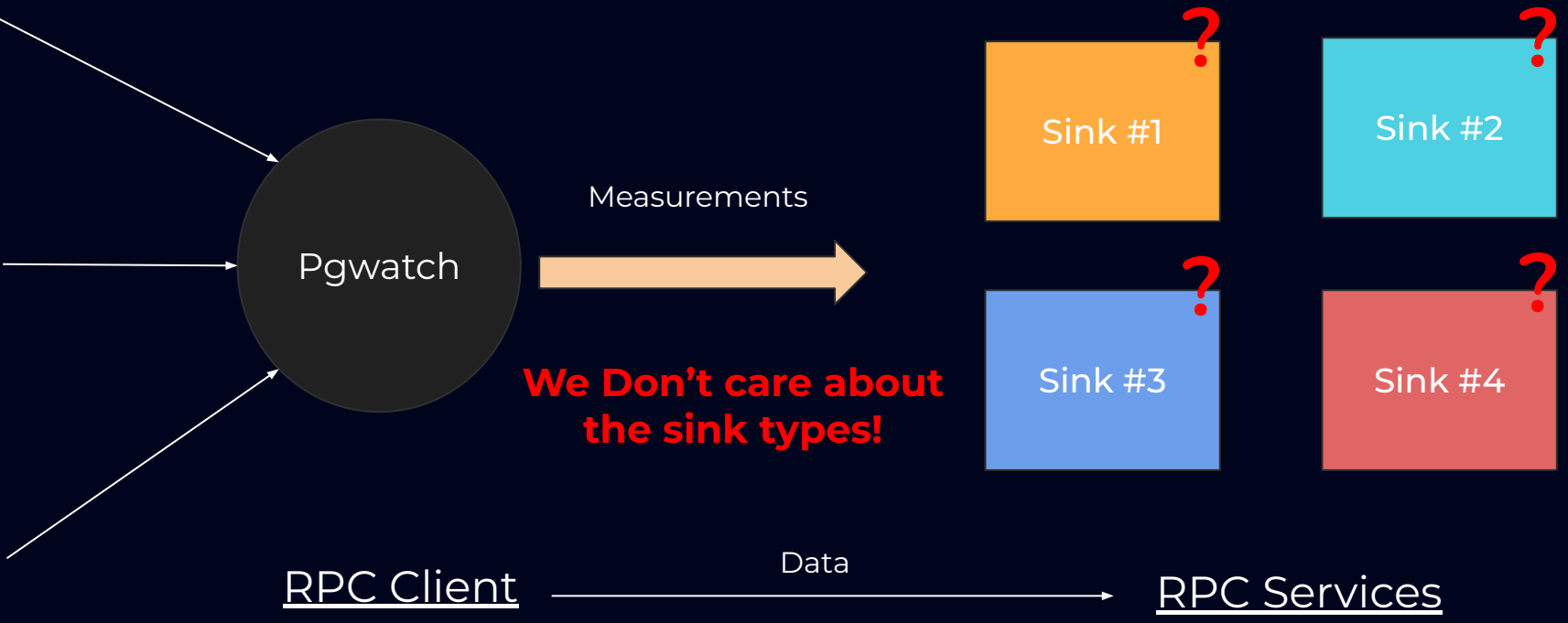
RPC Layer

PgWatch becomes Sink Agnostic!

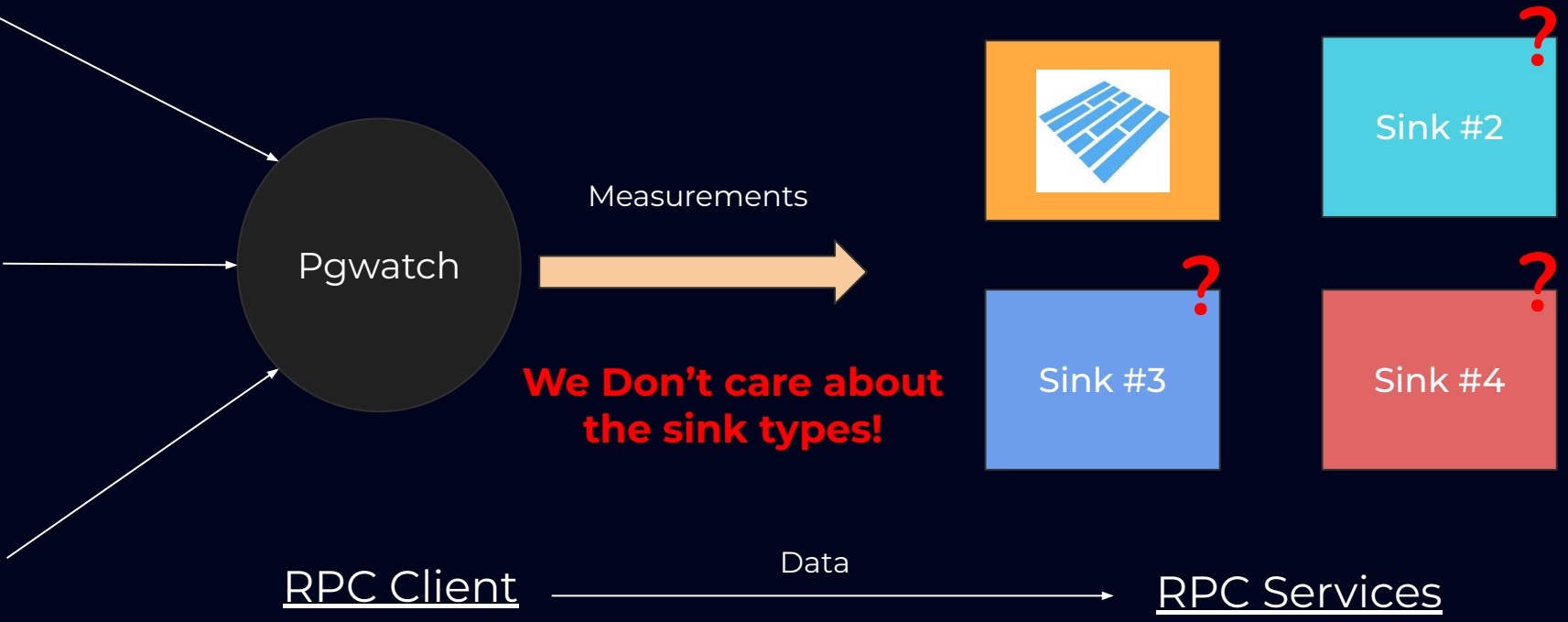
PgWatch becomes Sink Agnostic!

```
switch scheme {
case "jsonfile":
        w, err = NewJSONWriter(ctx, path)
case "postgres", "postgresql":
        w, err = NewPostgresWriter(ctx, s, opts, metricDefs)
case "prometheus":
        w, err = NewPrometheusWriter(ctx, path)
case "rpc":
        w, err = NewRPCWriter(ctx, path)
default:
        return nil, fmt.Errorf("unknown schema %s in sink URI %s", scheme, s)
}
```

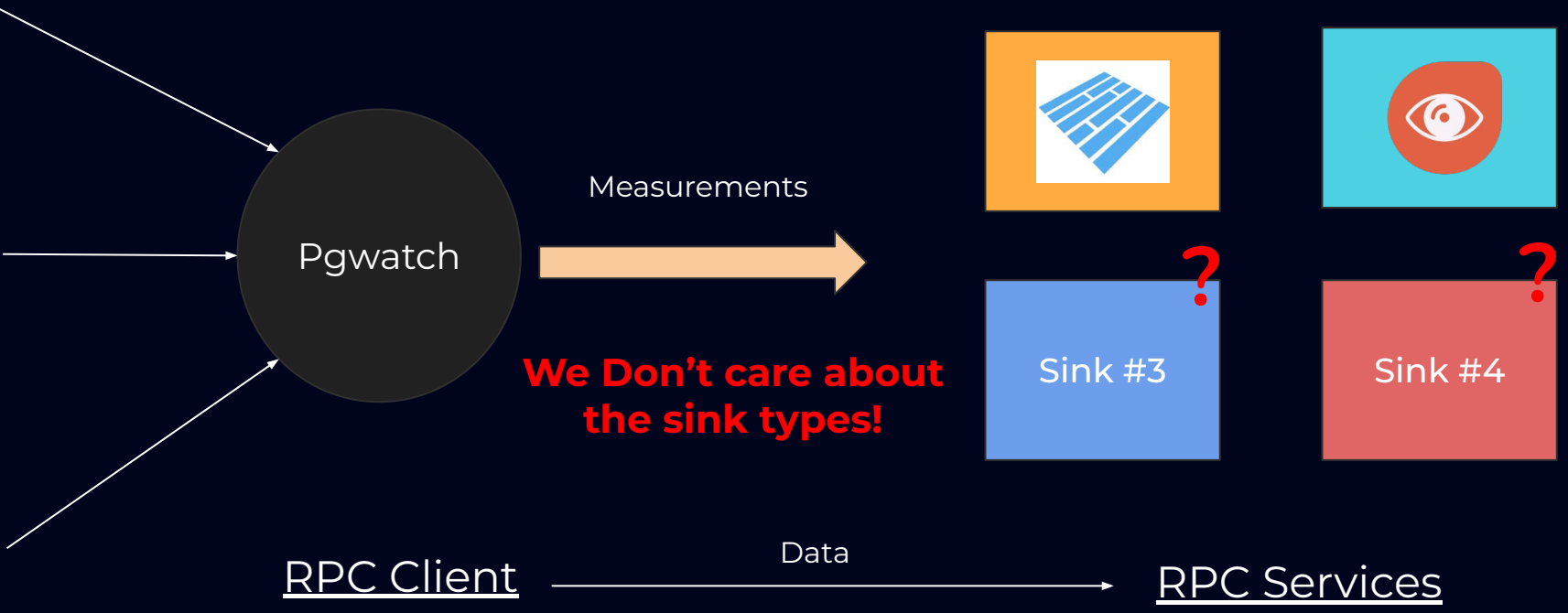
Remote Sinks Architecture



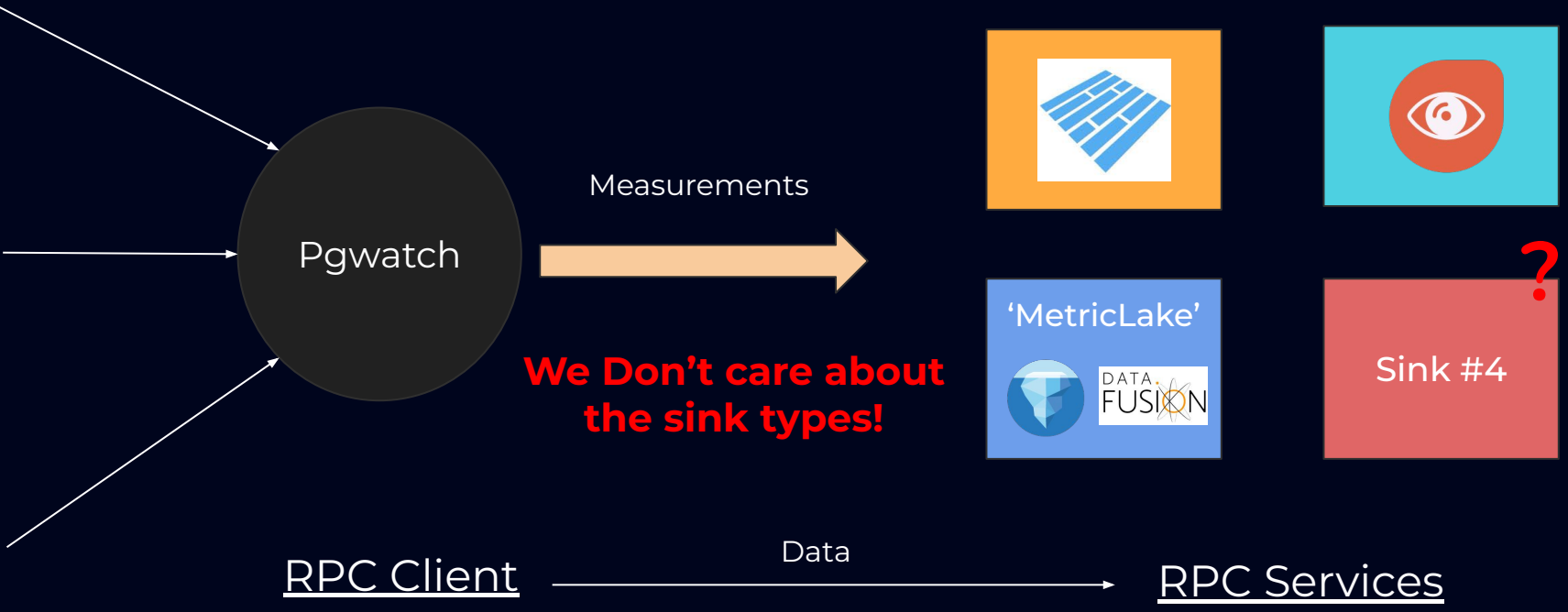
Remote Sinks Architecture



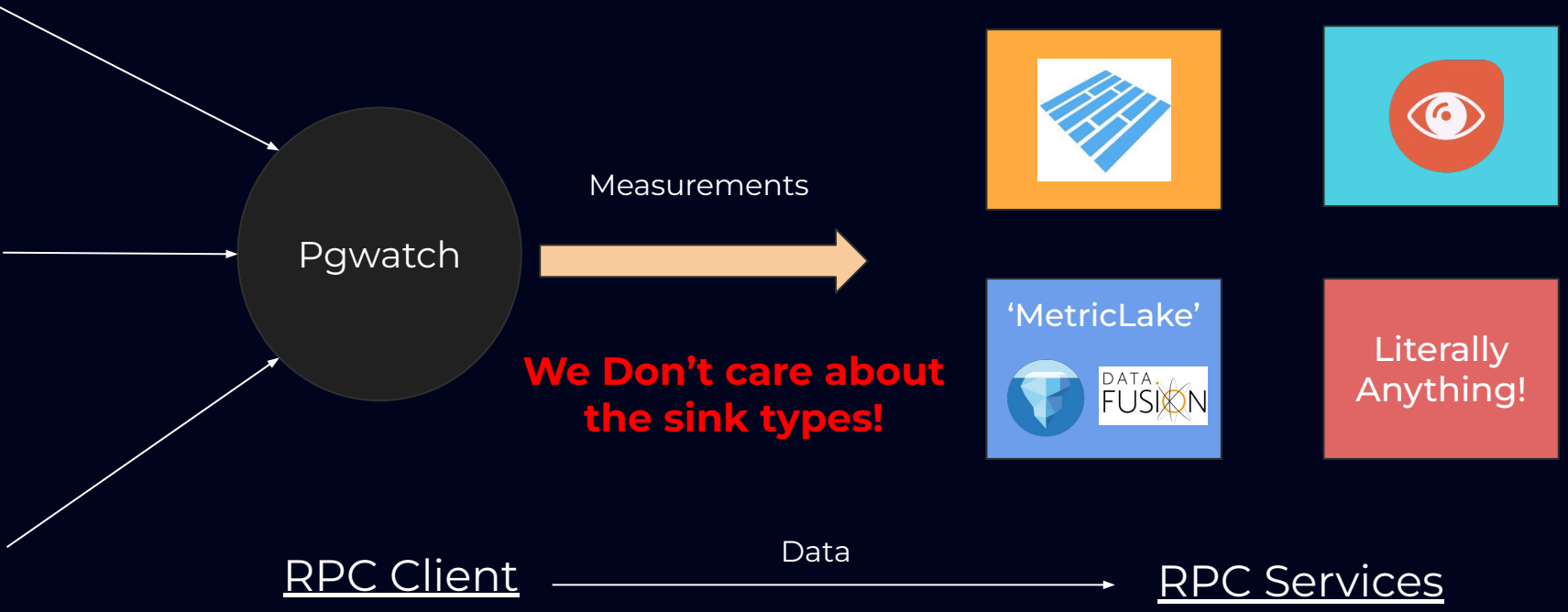
Remote Sinks Architecture



Remote Sinks Architecture



Remote Sinks Architecture



Sinks become truly Decoupled

```
if err := rw.client.Call("Receiver.UpdateMeasurements", &msg, &logMsg); err != nil {
        return err
}
```

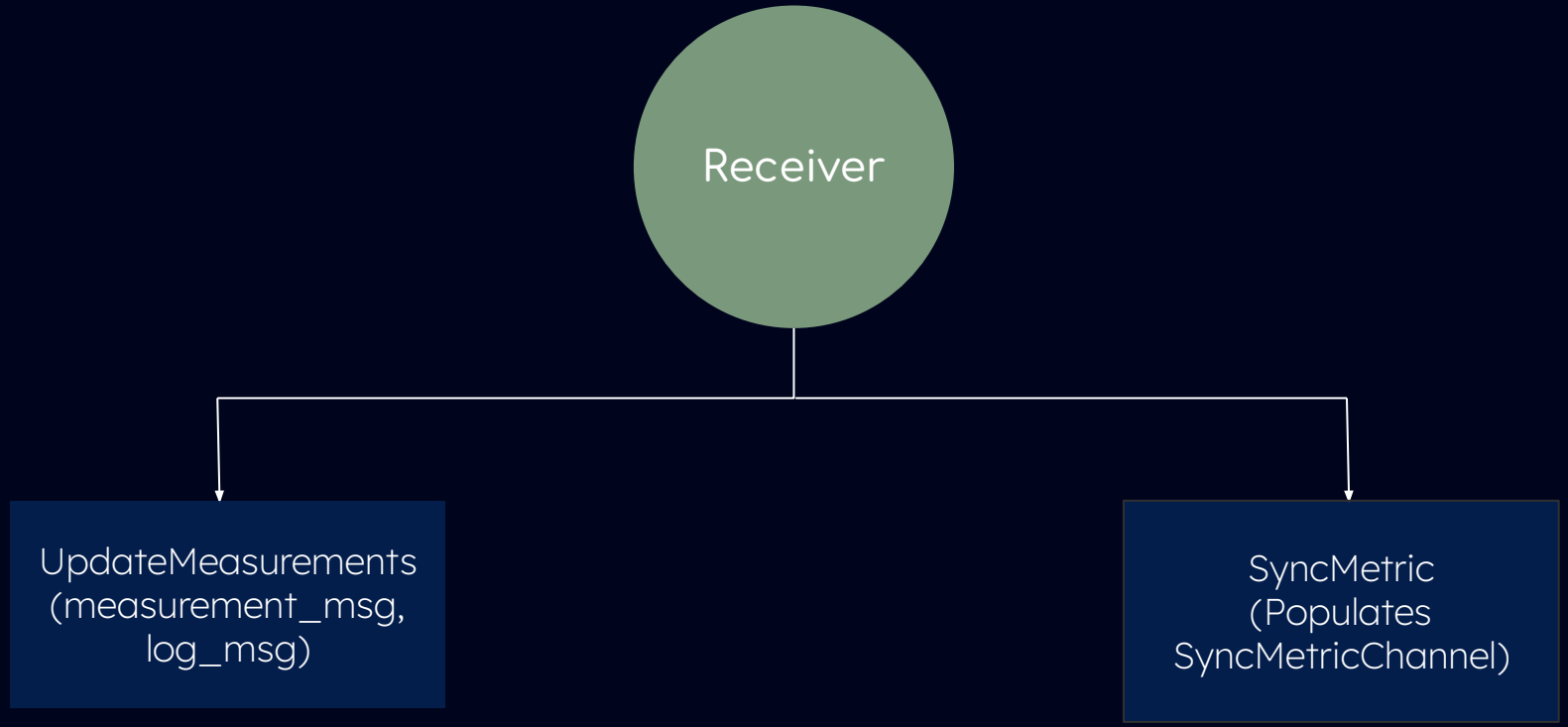
Easier To implement your own Sinks!

```go
type Receiver interface {
        UpdateMeasurements(msg *api.MeasurementEnvelope, logMsg *string) error
        SyncMetric(syncReq *api.RPCSyncRequest, logMsg *string) error
}
```

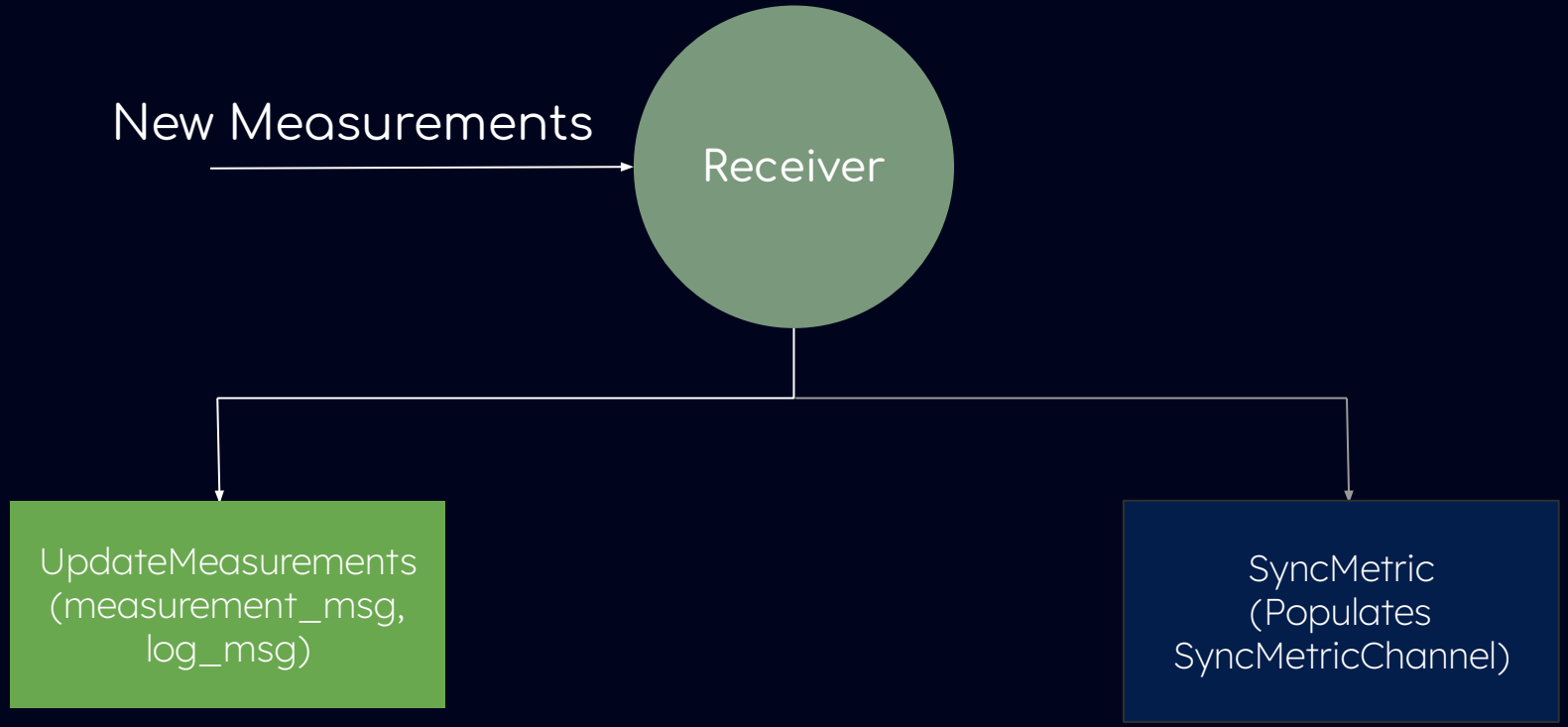
Quick Recap

- Pgwatch - Easy to use monitoring solution for PostgreSQL
- Sources, Metrics & Sinks - 3 primary pillars behind pgwatch
- Sinks - To Store Measurements extracted by pgwatch
- Remote Sinks - A easy to use interface for implementing your own custom sinks
- Remote Sinks - Provide loose coupling
- Remote Sinks - Make Pgwatch sink agnostic

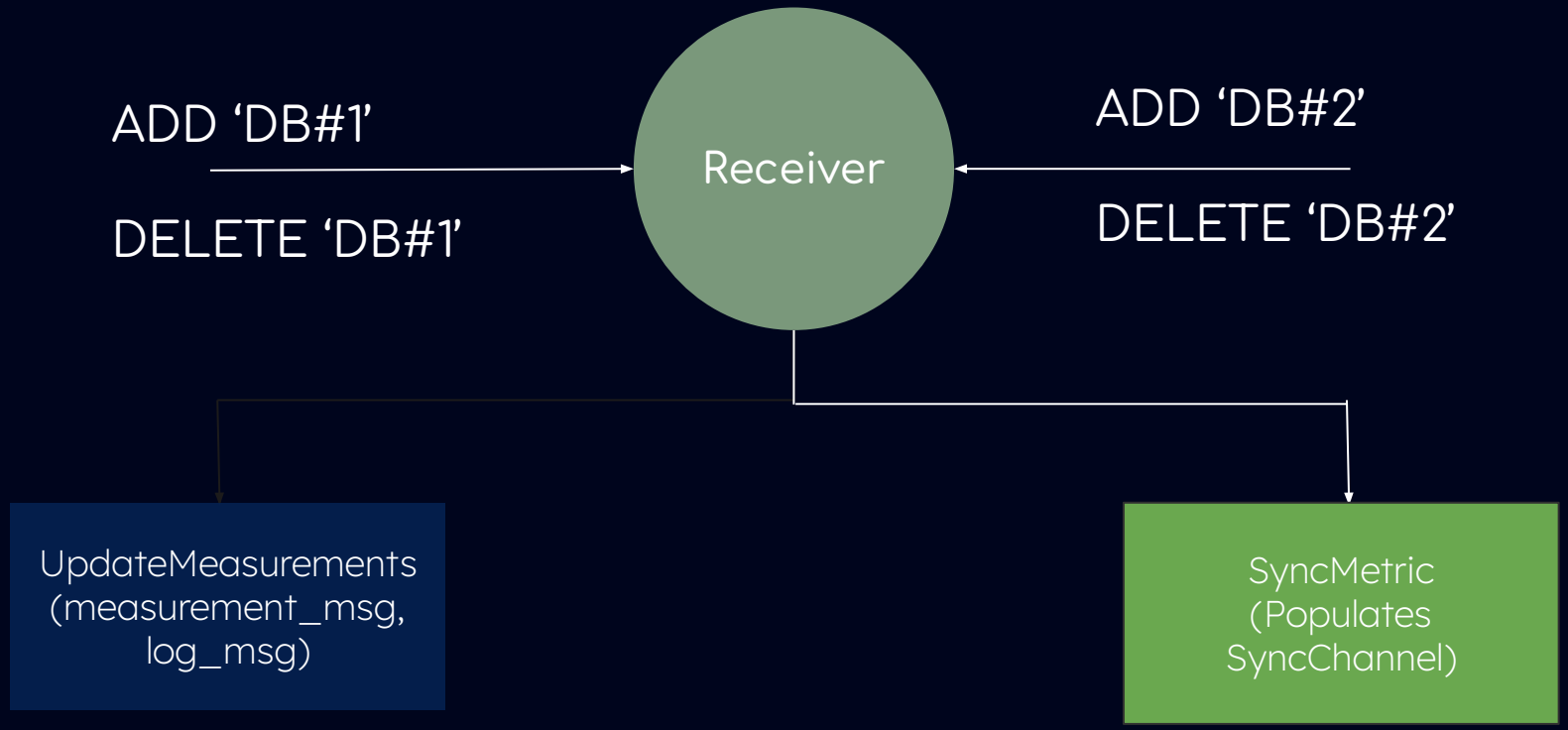
Developing Custom Sinks



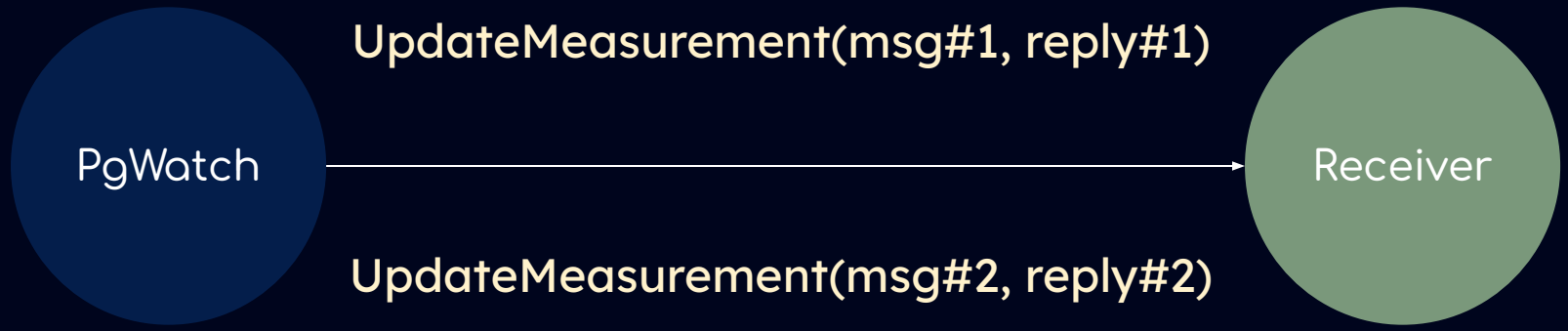
Developing Custom Sinks



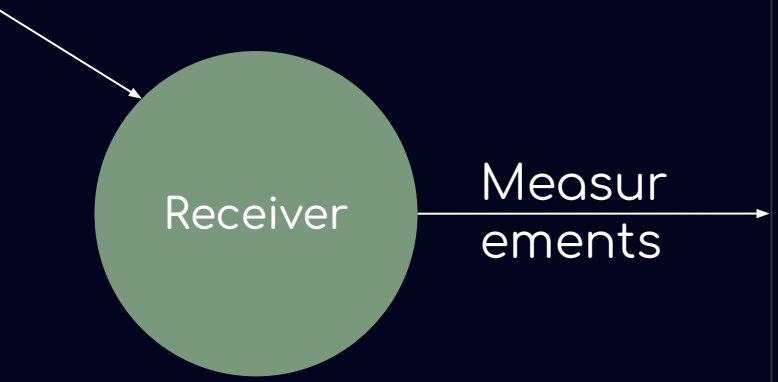
Developing Custom Sinks



Developing Custom Sinks

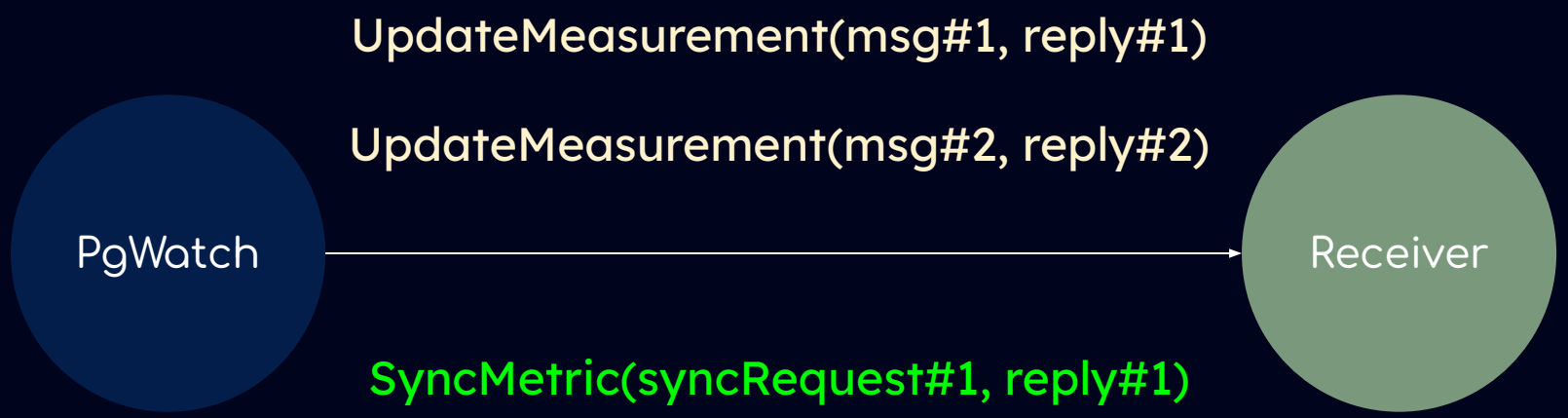


Developing Custom Sinks

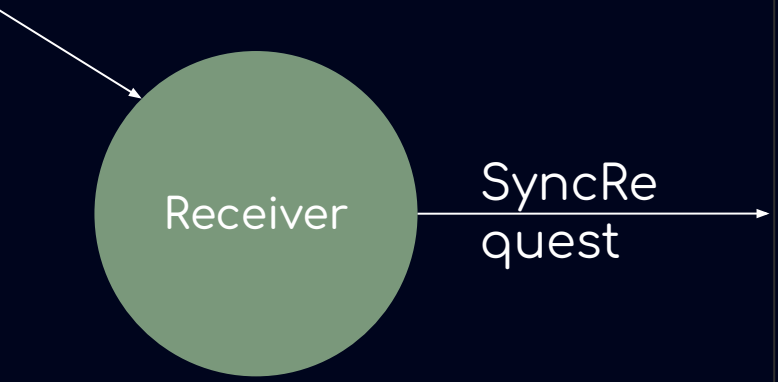



```
func (r *XYZReceiver) UpdateMeasurements(msg
api.MeasurementEnvelope, logMsg *string) error{

        // Create schema in XYZ
        err = XYZ.createTables(msg.DBName);

        If err != nil{
                *logMsg = “some error!!”
        }

.
.
.
}
```

Developing Custom Sinks



Developing Custom Sinks




```
func (r *KafkaProdReceiver) HandleSyncMetric() {

// Runs as in parallel
.
.
.
        req := <-r.SyncChannel

        switch req.Operation {
          case "DELETE":
            r.CloseConnectionForDB(req.DbName)
          case "ADD":
            r.AddTopicIfNotExists(req.DbName)
        }
}
```

https://github.com/destrex271/pgwatch3_rpc_server

Name	Last commit message	Last commit da...
..		
clickhouse_receiver	minor bug fixes and introduced batch inserts	2 weeks ago
csv_receiver	Update README.md	last month
kafka_prod_receiver	Updated readme and added comments	3 weeks ago
llama_receiver	LLM receiver readme	2 weeks ago
parquet_receiver	updated readme: added pkg link	2 weeks ago
s3_receiver	S3 receiver minor bug fixes	2 weeks ago
text_receiver	Added tests for text receiver (#21)	last month

Developing Custom Sinks

```go
type Receiver interface {
        UpdateMeasurements(msg *api.MeasurementEnvelope, logMsg *string) error
        SyncMetric(syncReq *api.RPCSyncRequest, logMsg *string) error
}
```

Developing Custom Sinks

```go
type Receiver interface {
        UpdateMeasurements(msg *api.MeasurementEnvelope, logMsg *string) error
        SyncMetric(syncReq *api.RPCSyncRequest, logMsg *string) error
}
```

```go
type KafkaProdReceiver struct {
        conn_regisrty map[string]*kafka.Conn
        uri           string
        auto_add      bool
        sinks.SyncMetricHandler
}
```

Developing Custom Sinks

```go
type KafkaProdReceiver struct {
        conn_regisrty map[string]*kafka.Conn
        uri           string
        auto_add      bool
        sinks.SyncMetricHandler
}
```

```go
func (r *KafkaProdReceiver) UpdateMeasurements(msg *api.MeasurementEnvelope, logMsg *string) error {
        // Kafka Recv
        // Get connection for database topic
        conn := r.conn_regisrty[msg.DBName]
        .
        .
        .
```

Developing Custom Sinks

```go
type KafkaProdReceiver struct {
        conn_regisrty map[string]*kafka.Conn
        uri           string
        auto_add      bool
        sinks.SyncMetricHandler
}
```

```go
func (r *KafkaProdReceiver) HandleSyncMetric() {
        req := <-r.SyncChannel
        switch req.Operation {
        case "DELETE":
                r.CloseConnectionForDB(req.DbName)
        case "ADD":
                r.AddTopicIfNotExists(req.DbName)
        }
}
```

```go
func (r *KafkaProdReceiver) UpdateMeasurements(msg *api.MeasurementEnvelope, logMsg *string) error {
        // Kafka Recv
        // Get connection for database topic
        conn := r.conn_regisrty[msg.DBName]
        .
        .
        .
```

```
[akshat arch] - [~/gsoc/pgwatch3] - [Tue Oct 08, 22:14]
[$] <git:(master*)> go run ./cmd/pgwatch --sources=postgres://postgres:postgres@localhost:5432/postgres --sink=rpc://127.0.0.1:8000 --sink=postgres://postgres:postgres@localhost:5432/pgwatch --sink=rpc://127.0.0.1:8001
2024-10-08 22:14:26.825 [INFO] [sink:rpc://127.0.0.1:8000] measurements sink actviated
2024-10-08 22:14:26.837 [INFO] [sink:postgres://postgres:postgres@localhost:5432/pgwatch] initialising the measurement database...
2024-10-08 22:14:26.845 [INFO] [sink:postgres://postgres:postgres@localhost:5432/pgwatch] measurements sink actviated
2024-10-08 22:14:26.846 [INFO] [sink:rpc://127.0.0.1:8001] measurements sink actviated
2024-10-08 22:14:26.849 [INFO] [metrics:75] [sources:2] host info refreshed
2024-10-08 22:14:26.850 [ERROR] Empty Database
2024-10-08 22:14:26.851 [ERROR] Empty Database
2024-10-08 22:14:26.876 [INFO] Connect OK. Version: PostgreSQL 17.0 (Debian 17.0-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit (in recovery: false)
2024-10-08 22:14:26.876 [INFO] Trying to create helper functions if missing for "db2"...
2024-10-08 22:14:26.889 [INFO] [source:db2] [metric:wal] [interval:60] starting gatherer
2024-10-08 22:14:26.892 [INFO] [source:db2] [metric:db_size] [interval:300] starting gatherer
2024-10-08 22:14:26.894 [INFO] [source:db2] [metric:db_stats] [interval:60] starting gatherer
2024-10-08 22:14:26.895 [INFO] [source:db2] [metric:wal] [rows:1] measurements fetched
2024-10-08 22:14:26.904 [INFO] [source:db2] [metric:db_stats] [rows:1] measurements fetched
2024-10-08 22:14:26.925 [INFO] Connect OK. Version: PostgreSQL 17.0 (Debian 17.0-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit (in recovery: false)
2024-10-08 22:14:26.925 [INFO] Trying to create helper functions if missing for "db1"...
2024-10-08 22:14:26.925 [INFO] [source:db2] [metric:db_size] [rows:1] measurements fetched
2024-10-08 22:14:26.937 [INFO] [source:db1] [metric:wal] [interval:60] starting gatherer
2024-10-08 22:14:26.938 [INFO] [source:db1] [metric:db_size] [interval:300] starting gatherer
2024-10-08 22:14:26.941 [INFO] [source:db1] [metric:db_stats] [interval:60] starting gatherer
2024-10-08 22:14:26.943 [INFO] [source:db1] [metric:wal] [rows:1] measurements fetched
2024-10-08 22:14:26.951 [INFO] [source:db1] [metric:db_stats] [rows:1] measurements fetched
2024-10-08 22:14:26.971 [INFO] [source:db1] [metric:db_size] [rows:1] measurements fetched
2024-10-08 22:14:27.133 [INFO] [sink:postgres] [rows:8] [db:pgwatch] [elapsed:13.520993ms] measurements written
```

Just specify sinks as -

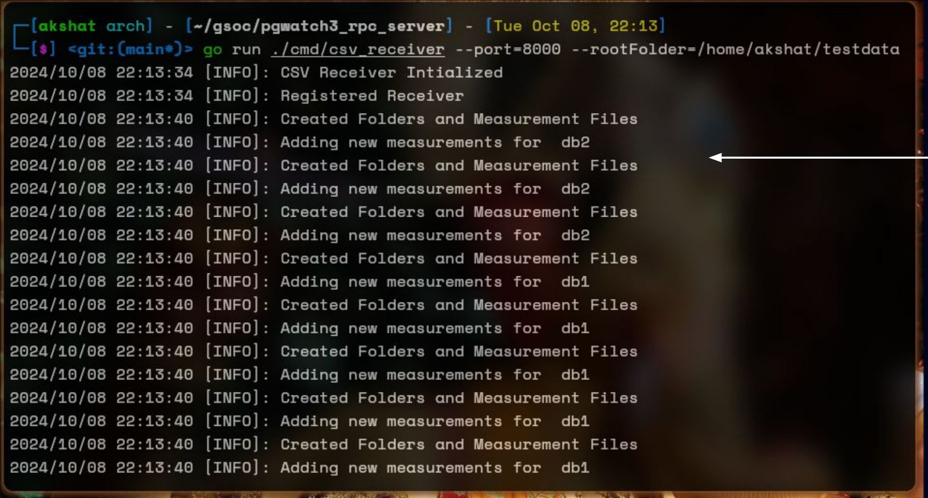
`- - sink=rpc://<host>:<port>`
`- - sink=rpc://<host>:<port>`
`- - sink=rpc://<host>:<port>`
`- - sink=rpc://<host>:<port>`

.
.
.

Setting Up Remote Sinks

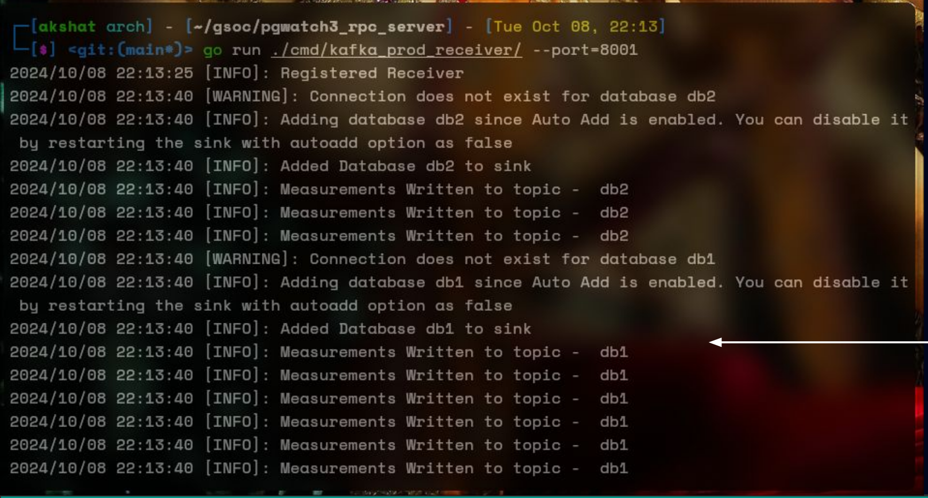
Using 2 receivers in parallel at the same time

CSV Receiver



```
┌[akshat arch] - [~/gsoc/pgwatch3_rpc_server] - [Tue Oct 08, 22:13]
└[$] <git:(main*)> go run ./cmd/csv_receiver --port=8000 --rootFolder=/home/akshat/testdata
2024/10/08 22:13:34 [INFO]: CSV Receiver Intialized
2024/10/08 22:13:34 [INFO]: Registered Receiver
2024/10/08 22:13:40 [INFO]: Created Folders and Measurement Files
2024/10/08 22:13:40 [INFO]: Adding new measurements for  db2
2024/10/08 22:13:40 [INFO]: Created Folders and Measurement Files
2024/10/08 22:13:40 [INFO]: Adding new measurements for  db2
2024/10/08 22:13:40 [INFO]: Created Folders and Measurement Files
2024/10/08 22:13:40 [INFO]: Adding new measurements for  db2
2024/10/08 22:13:40 [INFO]: Created Folders and Measurement Files
2024/10/08 22:13:40 [INFO]: Adding new measurements for  db1
2024/10/08 22:13:40 [INFO]: Created Folders and Measurement Files
2024/10/08 22:13:40 [INFO]: Adding new measurements for  db1
2024/10/08 22:13:40 [INFO]: Created Folders and Measurement Files
2024/10/08 22:13:40 [INFO]: Adding new measurements for  db1
2024/10/08 22:13:40 [INFO]: Created Folders and Measurement Files
2024/10/08 22:13:40 [INFO]: Adding new measurements for  db1
2024/10/08 22:13:40 [INFO]: Created Folders and Measurement Files
2024/10/08 22:13:40 [INFO]: Adding new measurements for  db1
```

Kafka Receiver



```
┌[akshat arch] - [~/gsoc/pgwatch3_rpc_server] - [Tue Oct 08, 22:13]
└[$] <git:(main*)> go run ./cmd/kafka_prod_receiver/ --port=8001
2024/10/08 22:13:25 [INFO]: Registered Receiver
2024/10/08 22:13:40 [WARNING]: Connection does not exist for database db2
2024/10/08 22:13:40 [INFO]: Adding database db2 since Auto Add is enabled. You can disable it
 by restarting the sink with autoadd option as false
2024/10/08 22:13:40 [INFO]: Added Database db2 to sink
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db2
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db2
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db2
2024/10/08 22:13:40 [WARNING]: Connection does not exist for database db1
2024/10/08 22:13:40 [INFO]: Adding database db1 since Auto Add is enabled. You can disable it
 by restarting the sink with autoadd option as false
2024/10/08 22:13:40 [INFO]: Added Database db1 to sink
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db1
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db1
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db1
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db1
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db1
2024/10/08 22:13:40 [INFO]: Measurements Written to topic -  db1
```

```
[akshat arch] - [~/gsoc/pgwatch3] - [Tue Oct 08, 22:14]
[*] <git:(master*)> go run ./cmd/pgwatch --sources=postgres://postgres:postgres@localhost:5432/postgres --sink=rpc://127.0.0.1:8000 --sink=postgres://postgres:postgres@localhost:5432/pgwatch --sink=rpc://127.0.0.1:8001
2024-10-08 22:14:26.825 [INFO] [sink:rpc://127.0.0.1:8000] measurements sink actviated
2024-10-08 22:14:26.837 [INFO] [sink:postgres://postgres:postgres@localhost:5432/pgwatch] initialising the measurement database...
2024-10-08 22:14:26.845 [INFO] [sink:postgres://postgres:postgres@localhost:5432/pgwatch] measurements sink actviated
2024-10-08 22:14:26.846 [INFO] [sink:rpc://127.0.0.1:8001] measurements sink actviated
2024-10-08 22:14:26.849 [INFO] [metrics:75] [sources:2] host info refreshed
2024-10-08 22:14:26.850 [ERROR] Empty Database
2024-10-08 22:14:26.851 [ERROR] Empty Database
2024-10-08 22:14:26.876 [INFO] Connect OK. Version: PostgreSQL 17.0 (Debian 17.0-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit (in recovery: false)
2024-10-08 22:14:26.876 [INFO] Trying to create helper functions if missing for "db2"...
2024-10-08 22:14:26.889 [INFO] [source:db2] [metric:wal] [interval:60] starting gatherer
2024-10-08 22:14:26.892 [INFO] [source:db2] [metric:db_size] [interval:300] starting gatherer
2024-10-08 22:14:26.894 [INFO] [source:db2] [metric:db_stats] [interval:60] starting gatherer
2024-10-08 22:14:26.895 [INFO] [source:db2] [metric:wal] [rows:1] measurements fetched
2024-10-08 22:14:26.904 [INFO] [source:db2] [metric:db_stats] [rows:1] measurements fetched
2024-10-08 22:14:26.925 [INFO] Connect OK. Version: PostgreSQL 17.0 (Debian 17.0-1.pgdg120+1) on x86_64-pc-linux-gnu, compiled by gcc (Debian 12.2.0-14) 12.2.0, 64-bit (in recovery: false)
2024-10-08 22:14:26.925 [INFO] Trying to create helper functions if missing for "db1"...
2024-10-08 22:14:26.925 [INFO] [source:db2] [metric:db_size] [rows:1] measurements fetched
2024-10-08 22:14:26.937 [INFO] [source:db1] [metric:wal] [interval:60] starting gatherer
2024-10-08 22:14:26.938 [INFO] [source:db1] [metric:db_size] [interval:300] starting gatherer
2024-10-08 22:14:26.941 [INFO] [source:db1] [metric:db_stats] [interval:60] starting gatherer
2024-10-08 22:14:26.943 [INFO] [source:db1] [metric:wal] [rows:1] measurements fetched
2024-10-08 22:14:26.951 [INFO] [source:db1] [metric:db_stats] [rows:1] measurements fetched
2024-10-08 22:14:26.971 [INFO] [source:db1] [metric:db_size] [rows:1] measurements fetched
2024-10-08 22:14:27.133 [INFO] [sink:postgres] [rows:8] [db:pgwatch] [elapsed:13.520993ms] measurements written
```

Sink Error Logs/Messages are also shared with pgwatch

Setting Up Remote Sinks

Live Demo!

Live Demo

Questions?!

Pavlo Golub

@pashagolub

@PavloGolub

Akshat Jaimini

@destrex271

@Kyllex5

akshatdev2711@gmail.com

Thank You & Let's Stay in Touch!

Do Share your Feedback with Us!