Invisible by Design

How Multigres keeps Postgres behaving like Postgres

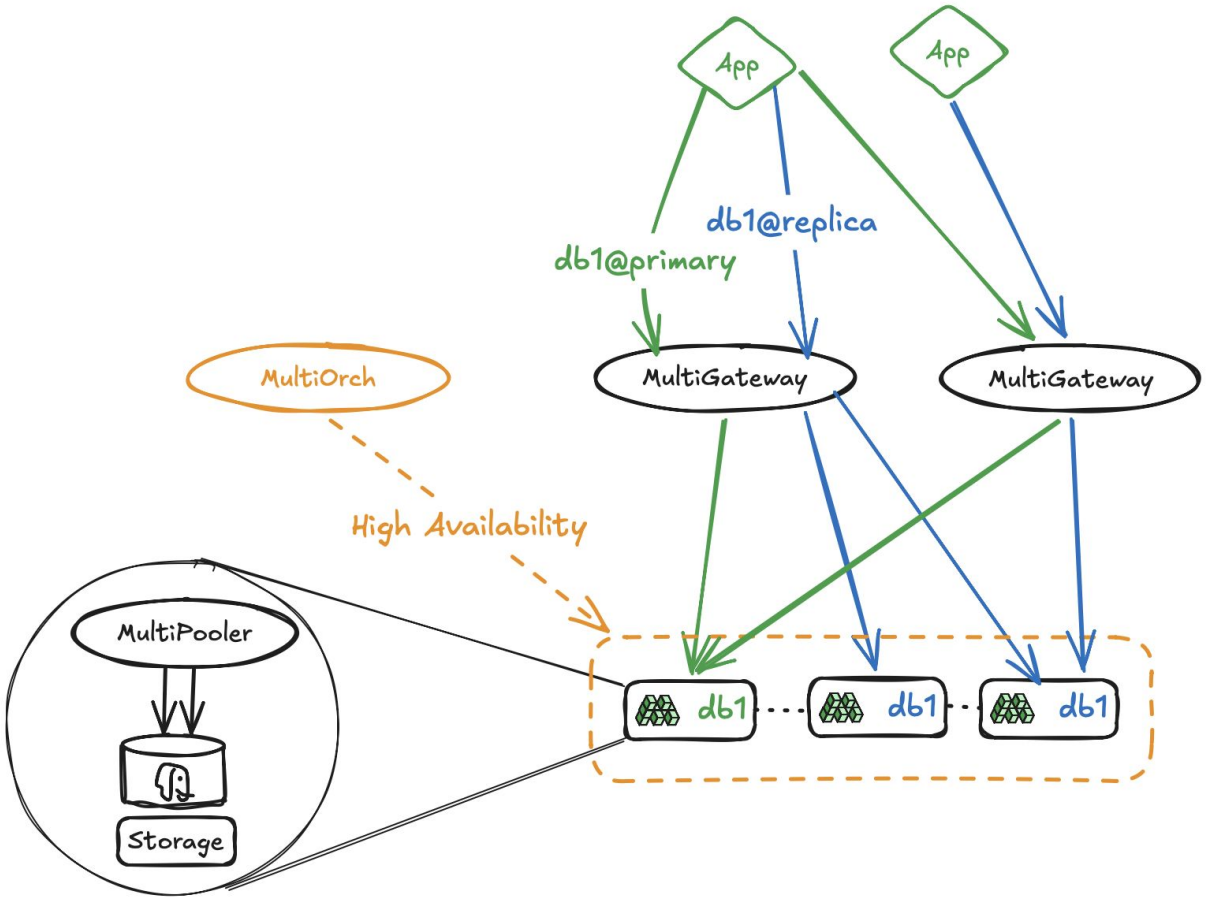
supabase

Manan Gupta // manan@supabase.io

October 2026

A client should never be able to tell it's talking to us *instead of real Postgres..*

The Architecture



Session state

Session State management

```
SET TIMEZONE = ‘UTC’
```

TIMEZONE	‘UTC’

```
SET CLIENT_ENCODING = ‘UTF8’
```

TIMEZONE	‘UTC’
CLIENT_ENCODING	‘UTF8’

```
SET TIMEZONE = ‘ASIA/KOLKATA’
```

TIMEZONE	‘ASIA/KOLKATA’
CLIENT_ENCODING	‘UTF8’

Some SET literals are multi-dimensional

```
-- backend A starts at German, YMD
SET datestyle = 'dmy';
```

PostgreSQL now reports German, DMY

```
-- Now change it again
SET datestyle = 'ISO';
```

PostgreSQL now reports ISO, DMY

Remembering the last value isn't enough for some multi-dimensional GUCs!

DateStyle: Use the ParameterStatus

One setting contains two independent components: style and date order.

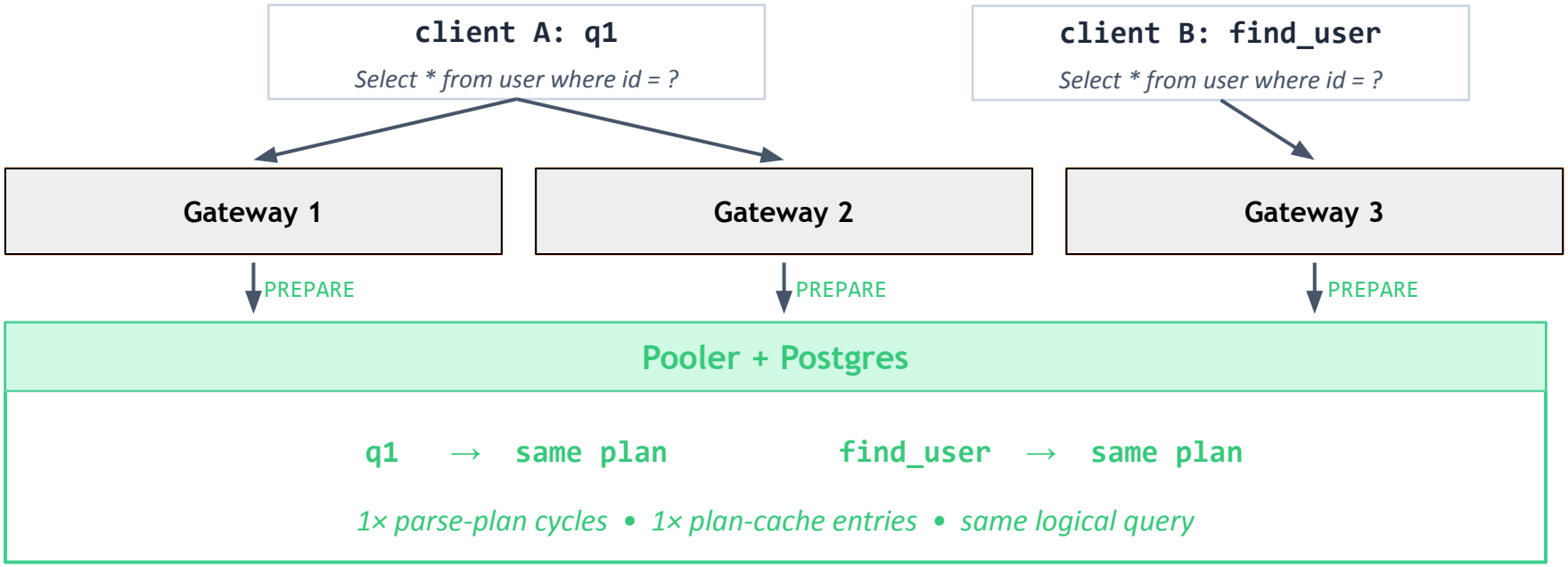


Track the ParameterStatus value PostgreSQL reports

DATESTYLE	German, DMY

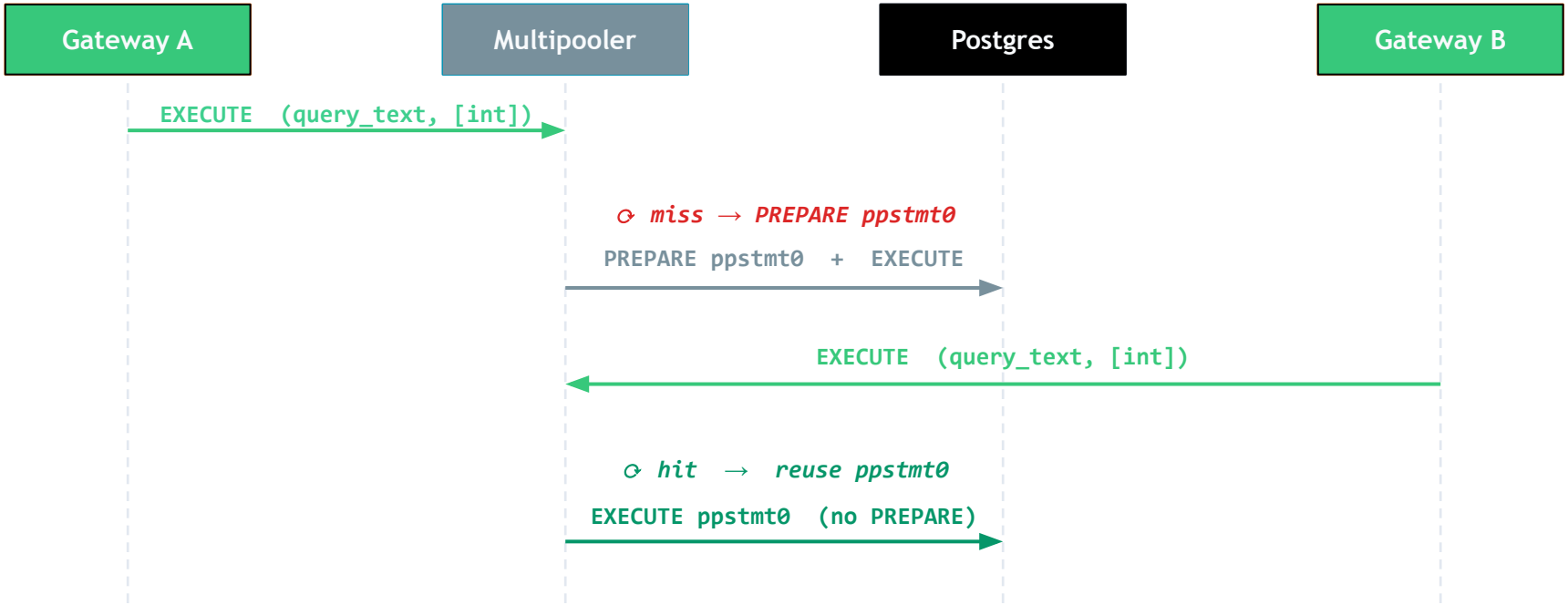
Stale prepared statements

The N-gateways and clients problem



Multiply by N gateways and the client reconnect rate.

The handoff



First time on a backend connection: PREPARE + EXECUTE. After that: EXECUTE only. Canonical name is stable across the whole pooler.

A fresh client Parse can inherit a stale plan

The pooler shares a backend statement by (query, parameter types).



DDL changes the schema

`0A000` `22P02` `42883`

Autocommit: detect and heal



Inside a transaction: re-Parse before failure

After an error, an in-place retry only produces “transaction is aborted”.



Close stale named statement → Parse against current schema

LISTEN / NOTIFY

LISTEN/NOTIFY breaks in transaction pooling

Transaction pooling can’t support LISTEN NOTIFY easily

1. **LISTEN foo** on backend #7
2. **Txn ends** pooler returns backend #7
3. **NOTIFY foo** delivered on backend #7
4. Nobody holds backend #7, client is gone
5. **× Notification orphaned!**

Use a reserved connection?

Reserved connections won't work

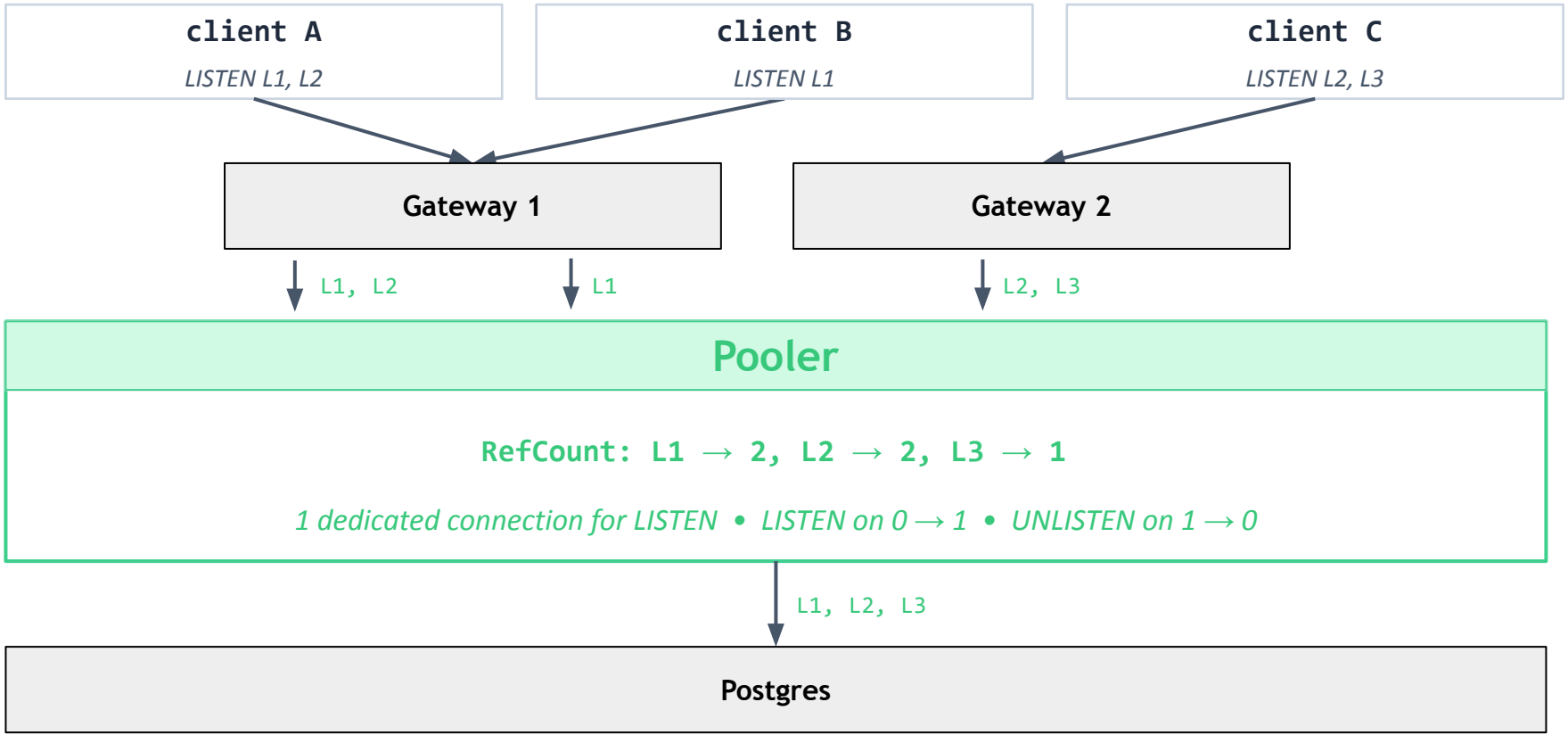
A LISTEN is long lived

Pin one backend per listener and it is held forever!

1,000 idle listeners = 1,000 reserved backends

If we can't pin it, and we can't drop it, what do we do?

Share the listener



COMMIT after an error

An aborted transaction still accepts COMMIT

```
BEGIN;
SELECT 1/0;  -- division by zero
```

The transaction is now aborted

Most commands fail with SQLSTATE 25P02.

```
COMMIT;
```

What should the client receive?

When COMMIT means ROLLBACK

```
BEGIN;
SELECT 1/0;  -- division by zero
COMMIT;
-- wire tag: ROLLBACK
```

No COMMIT error.
No commit.

"A client should never be able to tell it's talking to us instead of real Postgres, down to the surprising bits."

Query cancellation

The query cancel connection can land elsewhere

The connection is running through Gateway A

The client opens a new TCP connection to cancel it.

The CancelRequest arrives at Gateway B

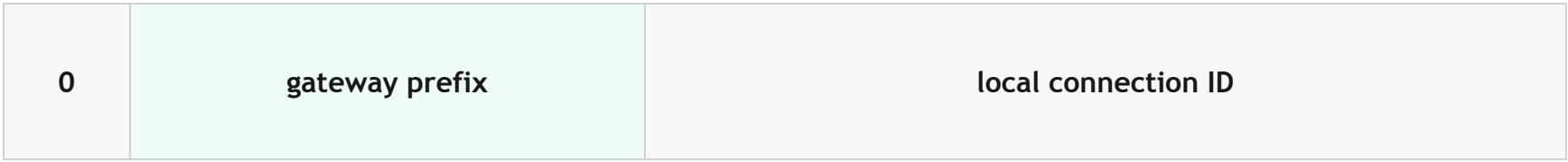
A physical PID cannot identify the logical session across the gateway fleet.

How do we send the cancellation request to the correct gateway?

Give the client session a virtual PID

A separate cancel connection has no stable backend to target.

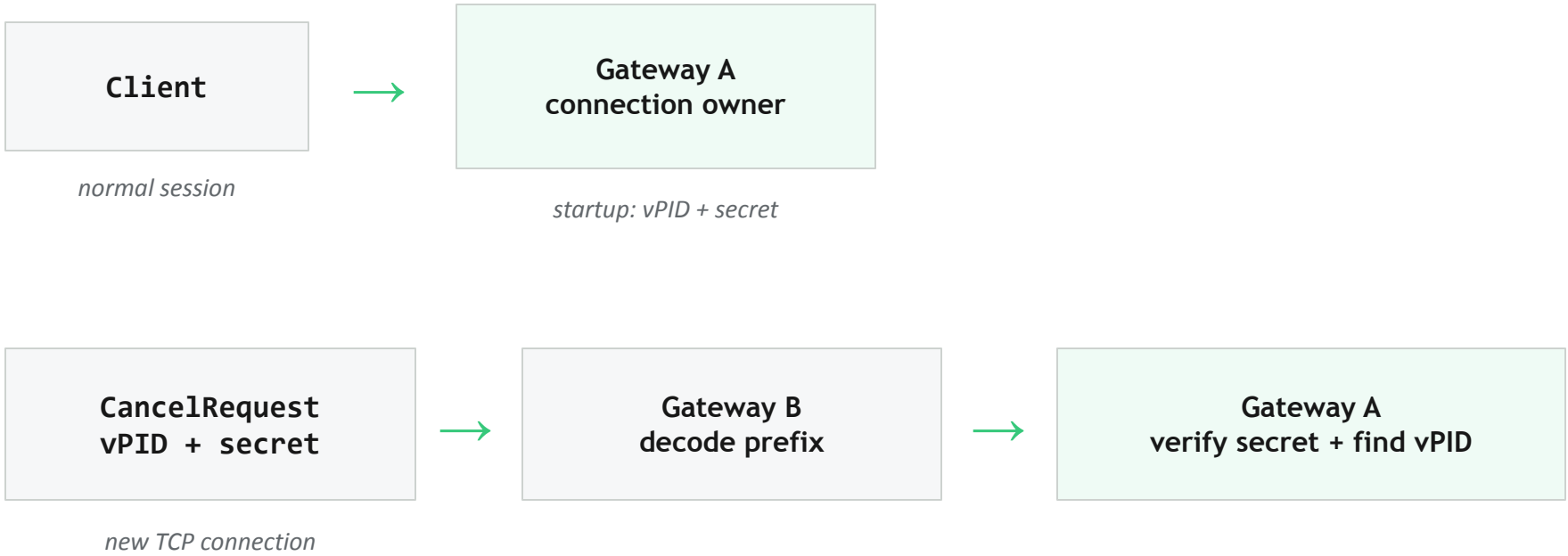
```
BackendKeyData  =  virtual PID + gateway-held secret
```




```
(prefix << LocalConnBits) | (localConnID & localConnMask)
```

Stable client identity. PostgreSQL backends need no VPID awareness.

Route cancellation to the owner



Owner cancels the active query context for the session's current work.

If your app can tell us apart from Postgres, file it as an issue!

manan@supabase.io

Manan Gupta • Multigres

github.com/multigres/multigres • multigres.com