# zenith storage system

Heikki Linnakangas

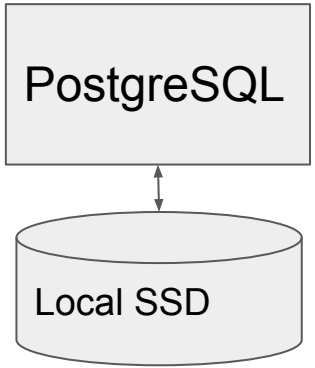
# What is zenith?

- New storage system for PostgreSQL
- Cloud service
- Startup

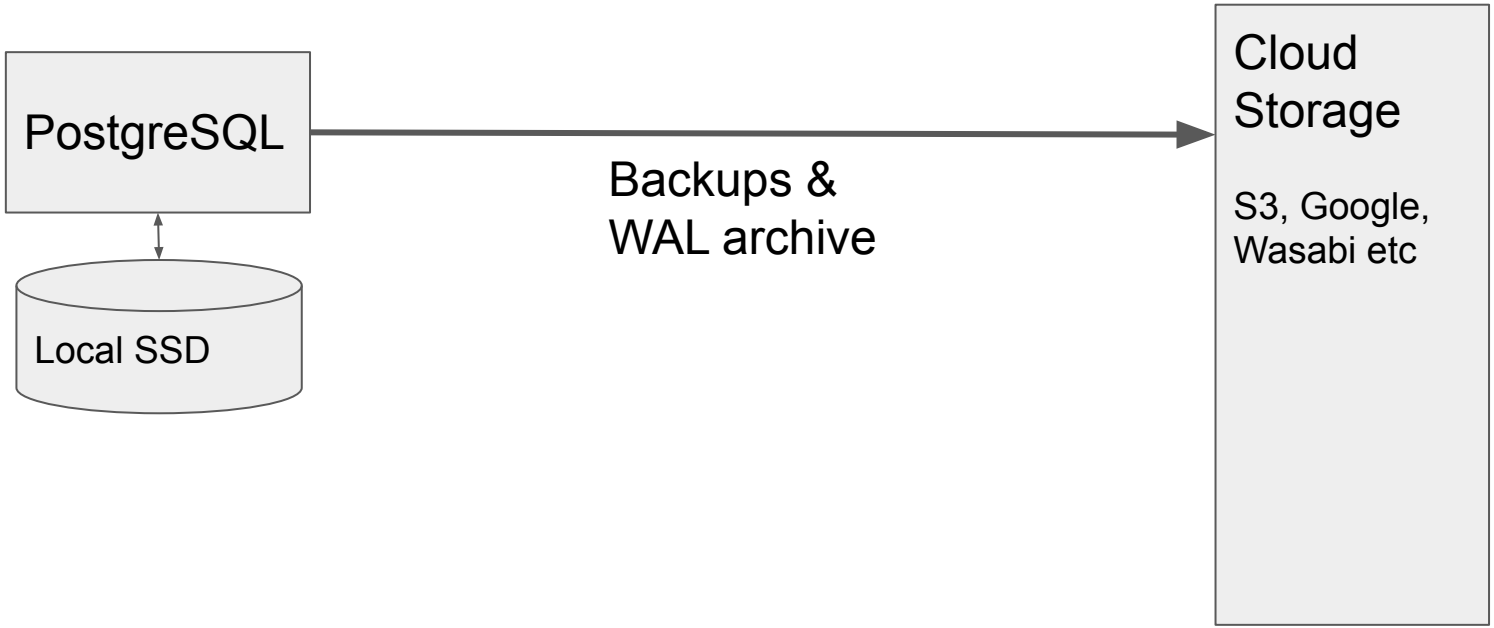
# Forewarning

- Work in progress
- Basic stuff works
- Architecture is sound
- But many of the features I’m describing have not actually been implemented yet.

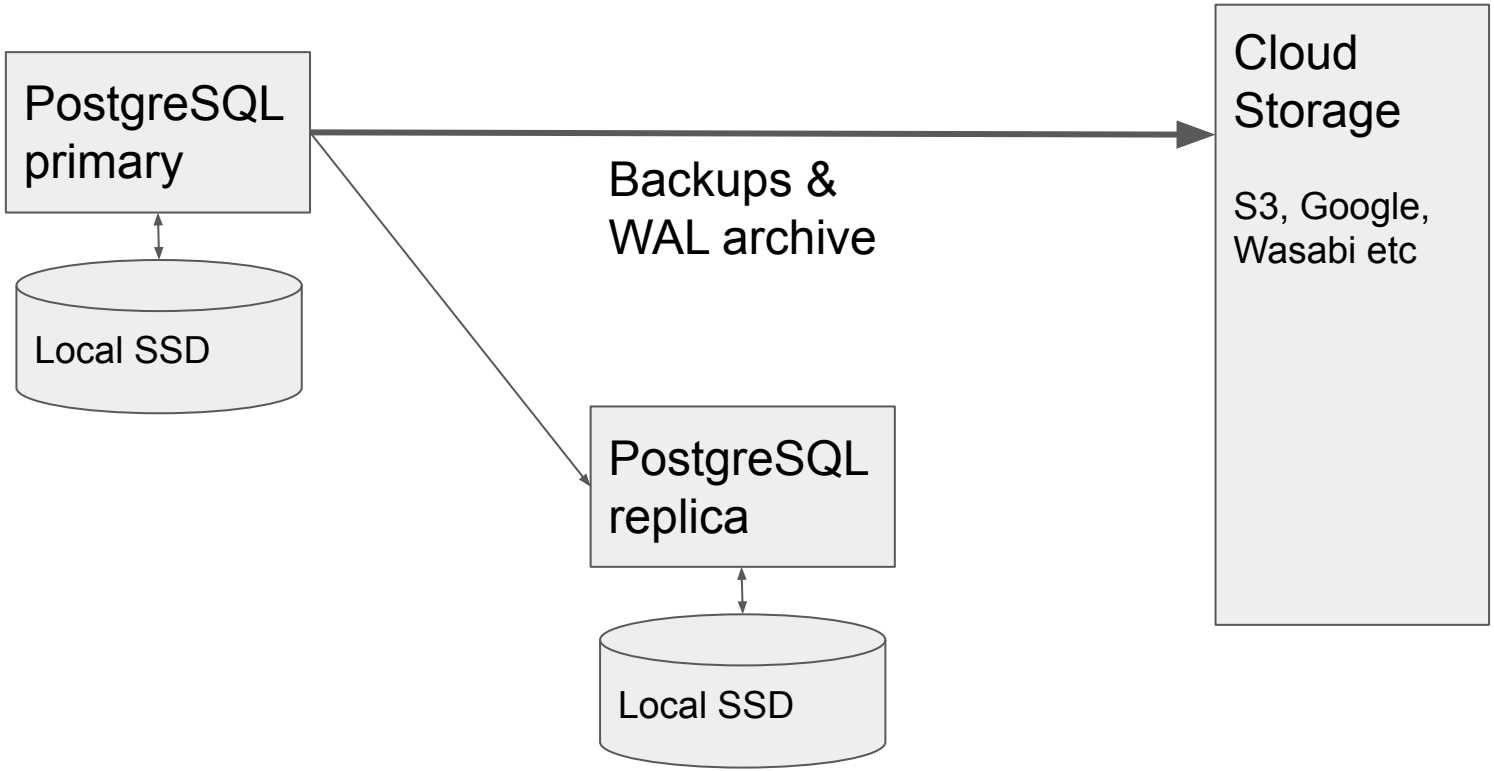
# Traditional PostgreSQL setup



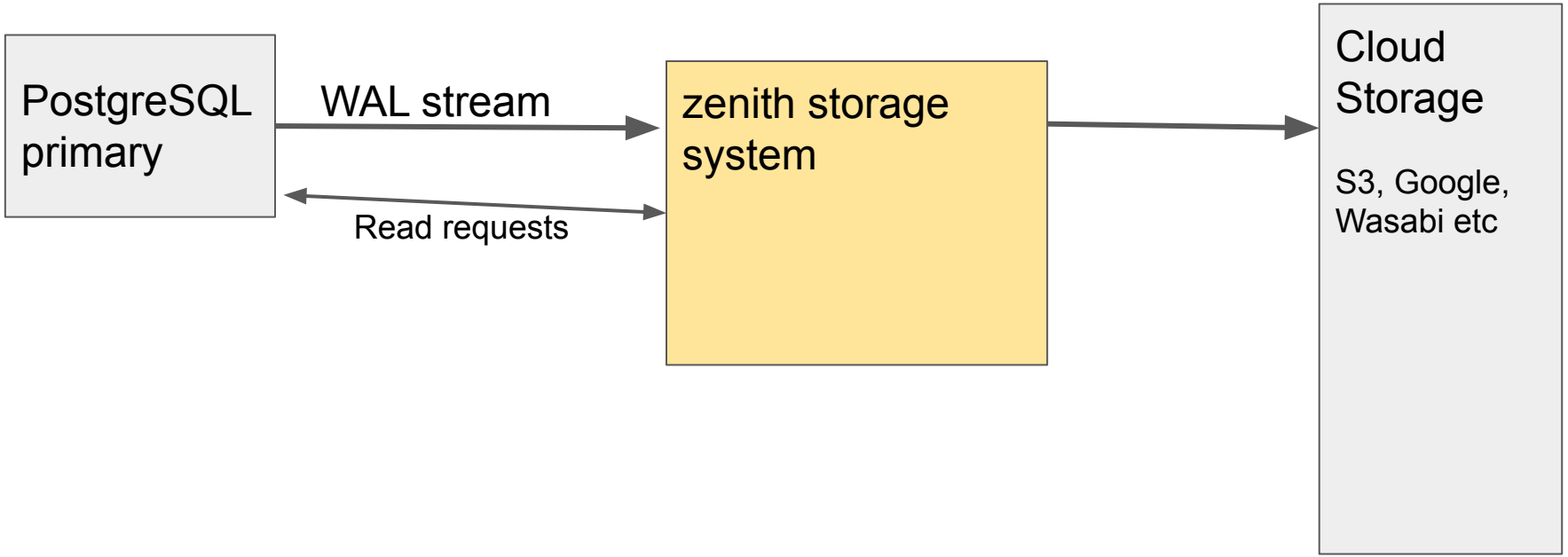
# Traditional PostgreSQL setup



# Traditional PostgreSQL setup



# Separate Compute and Storage



# Separating compute and storage

- Amazon Aurora
- PolarDB from Alibaba
- Microsoft Hyperscale
- Snowflake
- SingleStore

# zenith storage system

- Replaces local storage, backups and WAL archive
- Knows about PostgreSQL WAL and page format
- SAN on steroids

# No lengthy recovery needed

Traditional:

- Restore last backup, replay all WAL, can take hours

zenith:

- Start up **instantly**
- WAL is replayed on-demand, page at a time

# Setting up a read replica

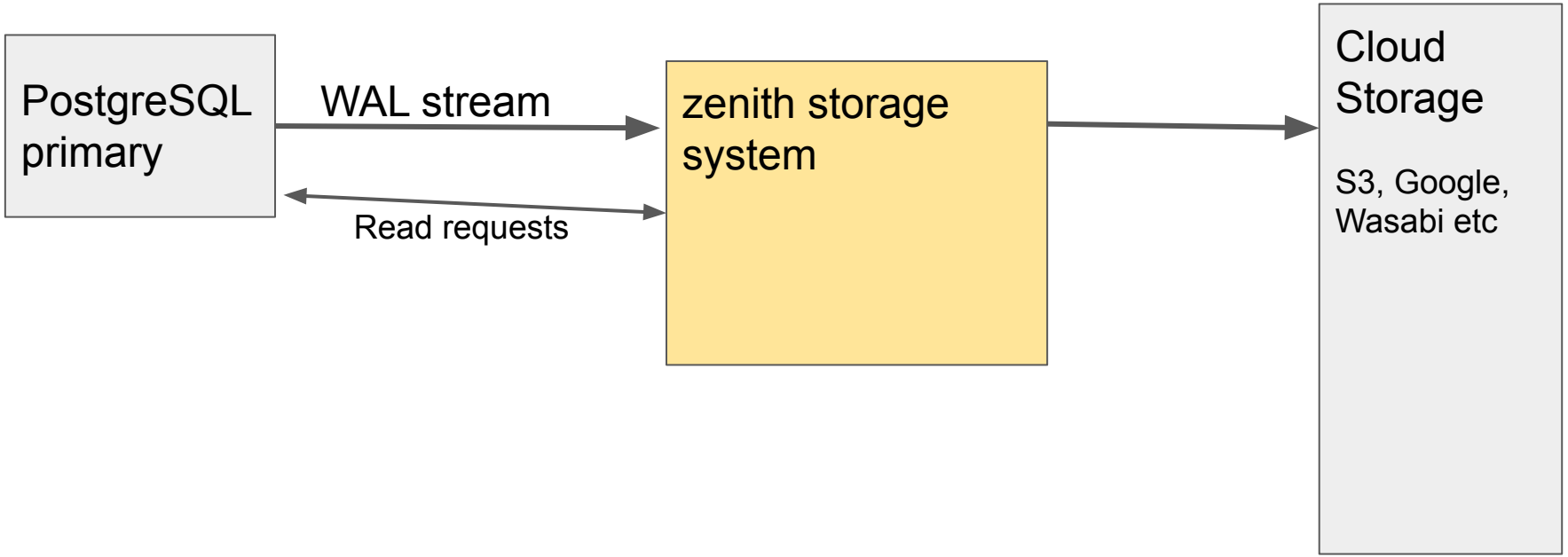
Traditional:

- Restore last backup, replay all WAL, can take hours
- Needs a fully copy of the data

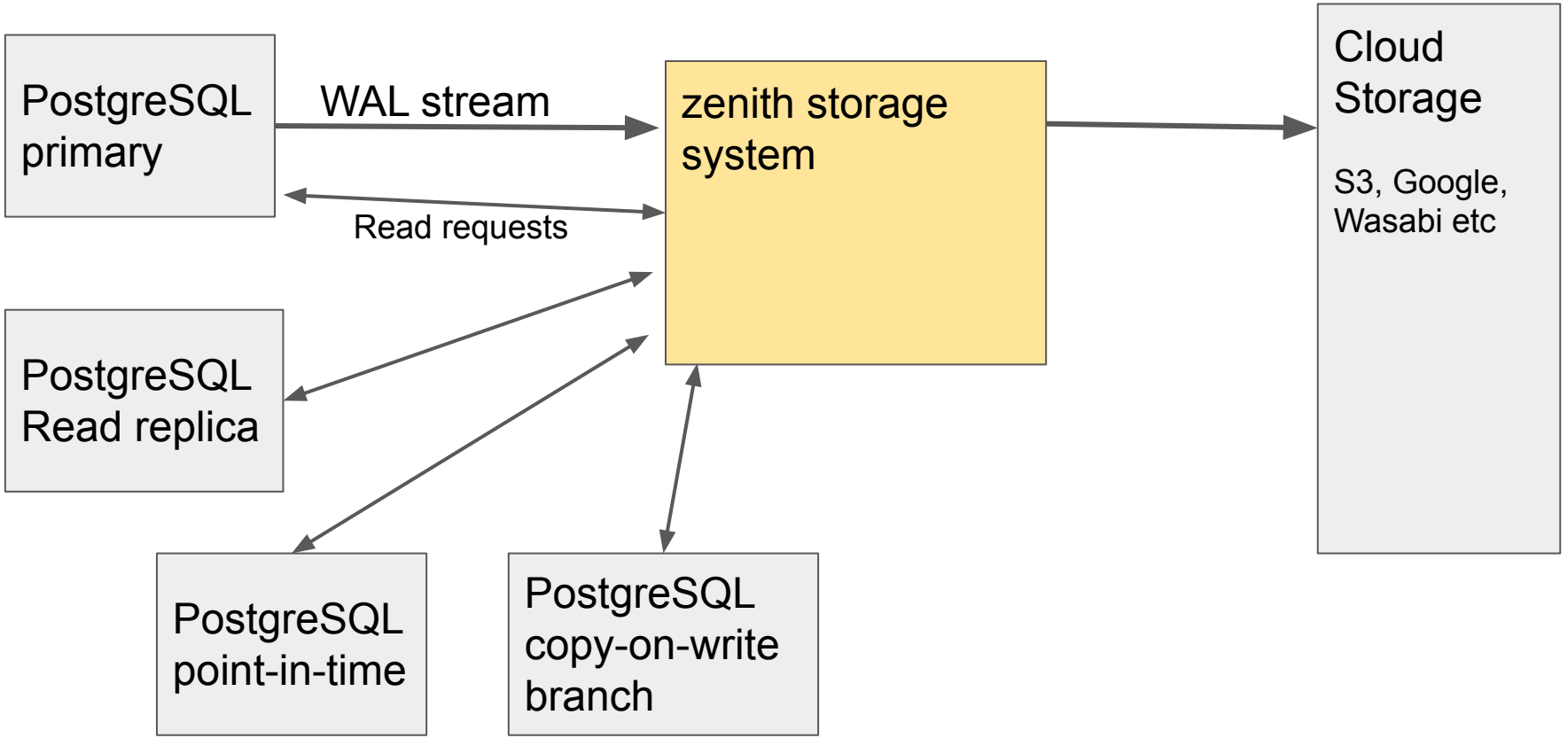
Zenith:

- Start up **instantly**
- Lightweight, shared storage

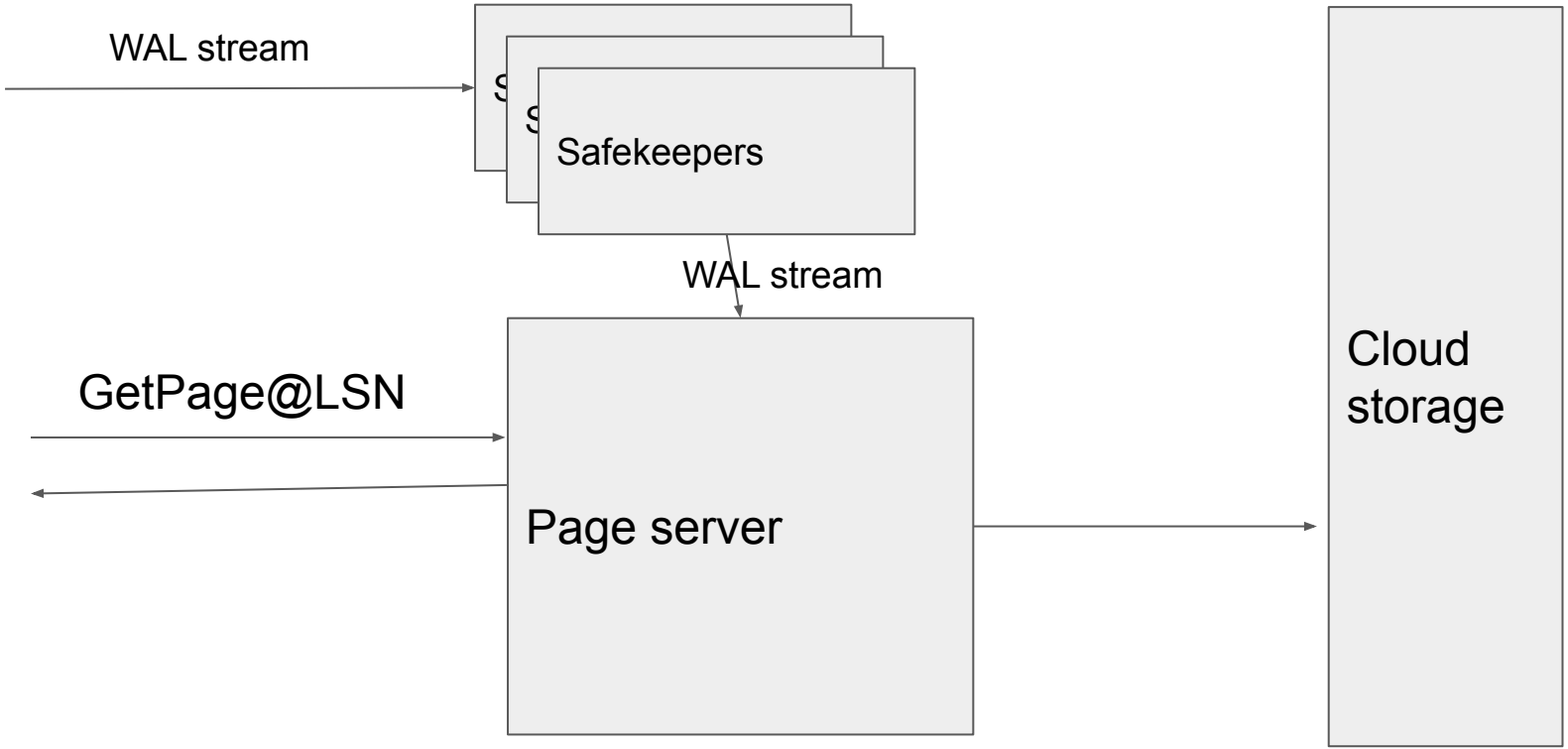
# Separate Compute and Storage



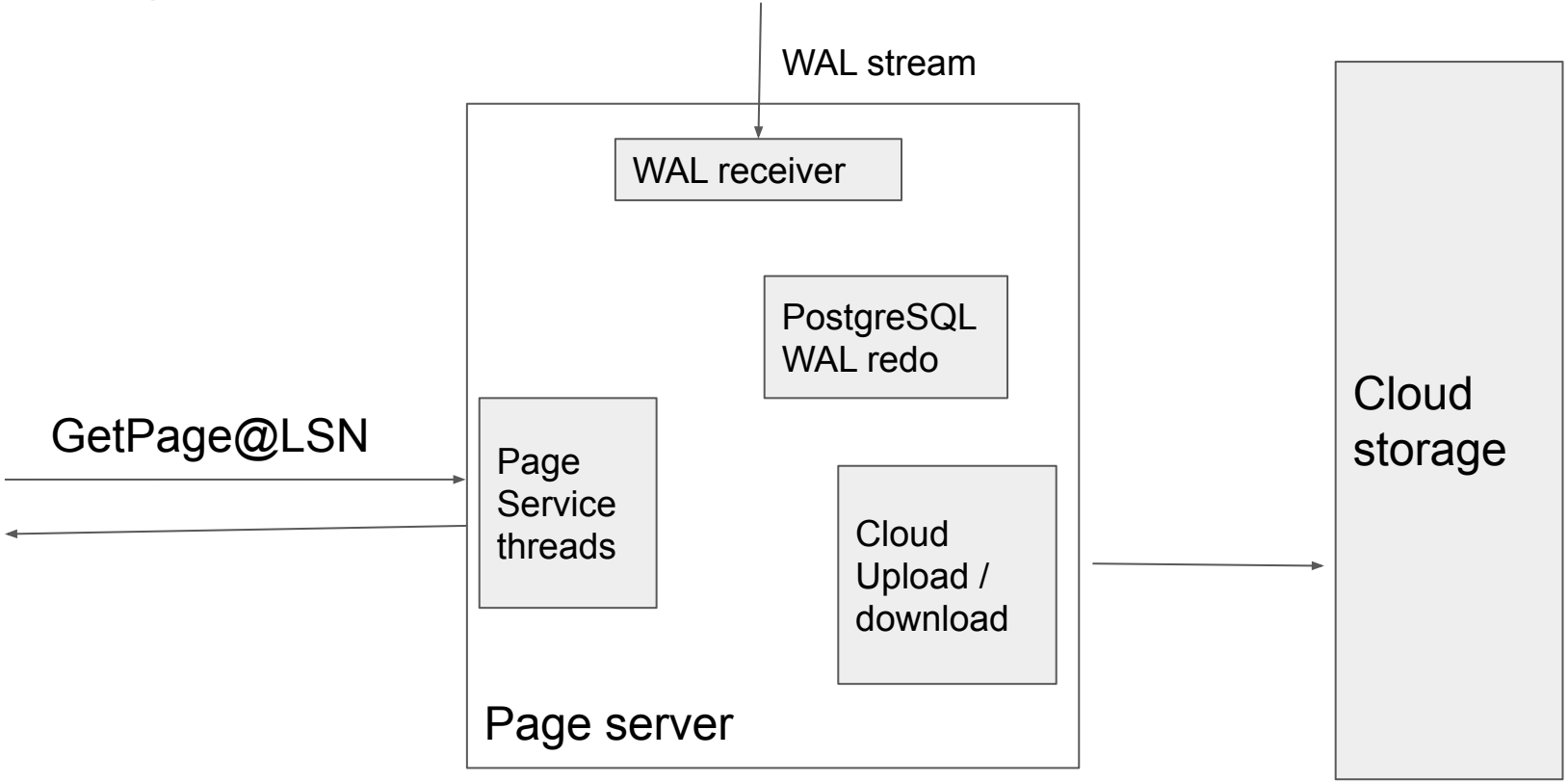
# Separate Compute and Storage



# Closer look


WAL stream
Safekeepers
WAL stream
GetPage@LSN
Page server
Cloud storage

# Page server internals



# Page server

- Heart of the storage system
- Receives WAL
- Parses it
- Writes it in a pre-processed format to local disk
- Uploads it to cloud storage

# Storage format

- Immutable files
- Similar to an LSM tree
- Compaction and garbage collection in the background
- Heuristics on when to store WAL records, and when to materialize an image of a page by replaying the WAL records

# GetPage@LSN

- Find the last image of the requested page, <= requested LSN, and any WAL records up to the LSN
- Replay the WAL records to reconstruct the requested version of the page

# Branching

- Like git branch, but for data
- Copy-on-Write
- CI/CD workflow:

    1. Create a branch of production database
    2. Run tests against the branch
    3. Drop the branch

# Current status

# Current Status

- All the code is on github
- Apache 2 license
- Storage code is written in Rust
- Passes PostgreSQL regression tests

# TODOs

- Read replicas that follow primary
- Sharding/scaling the storage
- Performance
- On-demand download from cloud storage
- Garbage collection

# PostgreSQL modifications

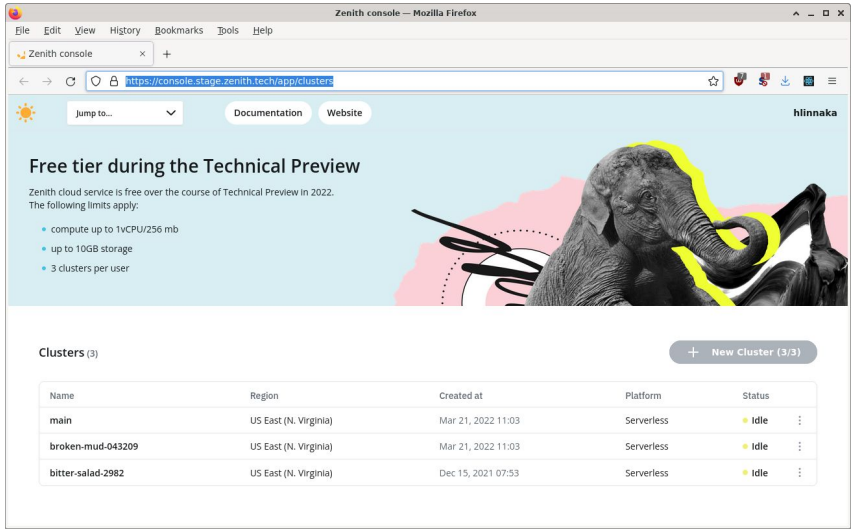
- Goal is to run unmodified PostgreSQL on zenith
  - Not quite there yet
- Smgr API changes
- Handling non-relation data like pg_xact
- Tracking LSN of last evicted page in buffer manager
- Unlogged tables

# Hosted Cloud service

- Serverless
- Web UI
- API for programmatic access
- Free tier

Try it: `psql -h start.zenith.tech`

Invite code: **nordic-2022**



# Thank you!

# Q & A

https://github.com/zenithdb/zenith/

Try it: `psql -h start.zenith.tech`
Invite code: **nordic-2022**

Feedback: heikki.linnakangas@iki.fi