# Garage

NGI POINTER

**Deuxfleurs Association**

`https://garagehq.deuxfleurs.fr/`

Matrix channel: `#garage:deuxfleurs.fr`

# Our objective at Deuxfleurs

**Promote self-hosting and small-scale hosting as an alternative to large cloud providers**

# Our objective at Deuxfleurs

**Promote self-hosting and small-scale hosting**
**as an alternative to large cloud providers**

Why is it hard?

# Our objective at Deuxfleurs

**Promote self-hosting and small-scale hosting as an alternative to large cloud providers**

Why is it hard?

**Resilience**

(we want good uptime/availability with low supervision)

# How to be resilient (the hard way)

Entreprise-grade systems typically employ:

- RAID
- Redundant power grid + UPS
- Redundant Internet connections
- Low-latency links
- ...

→ it's costly and only worth it at DC scale

# How to be resilient (the **cheap** way)

Instead, we use:

- Commodity hardware (e.g. old desktop PCs)

# How to be resilient (the **cheap** way)

# How to be resilient (the **cheap** way)

# How to be resilient (the **cheap** way)

Instead, we use:

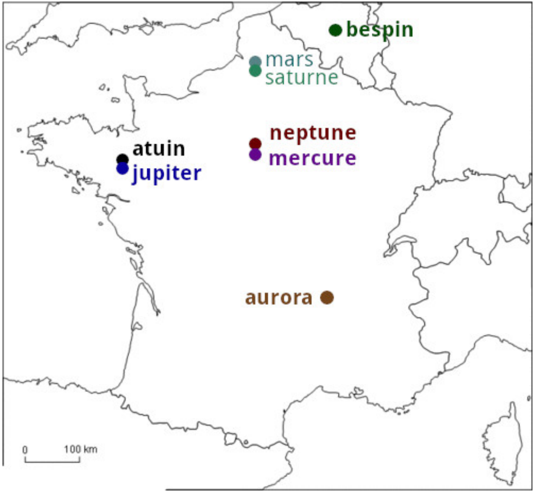
- Commodity hardware (e.g. old desktop PCs)
- Commodity Internet (e.g. FTTB, FTTH) and power grid

# How to be resilient (the **cheap** way)

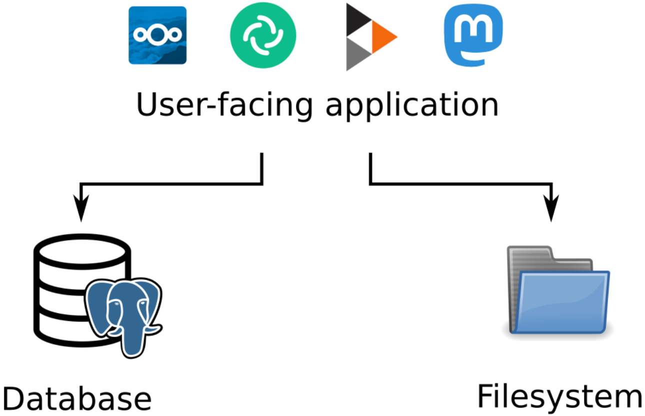
Instead, we use:

- Commodity hardware (e.g. old desktop PCs)
- Commodity Internet (e.g. FTTB, FTTH) and power grid
- **Geographical redundancy** (multi-site replication)

# How to be resilient (the **cheap** way)



# How to make this happen

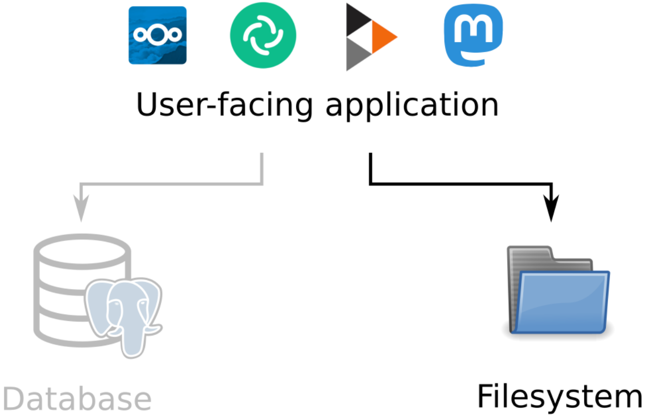


User-facing application

Database

Filesystem

# How to make this happen

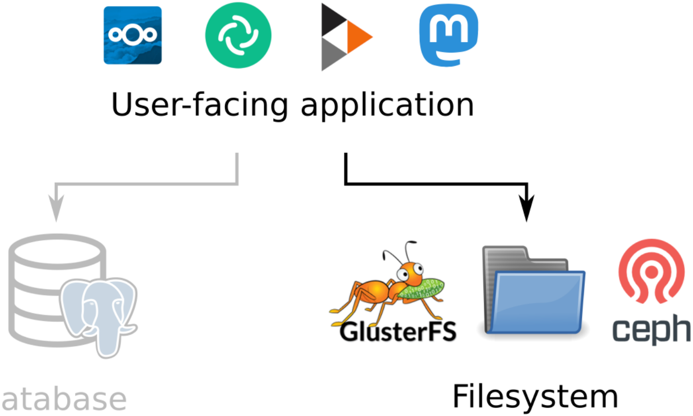


User-facing application

Database

Filesystem

# How to make this happen



User-facing application

Database

Filesystem

# Distributed file systems are slow

File systems are complex, for example:

- Concurrent modification by several processes
- Folder hierarchies
- Other requirements of the POSIX spec

Coordination in a distributed system is costly

Costs explode with commodity hardware / Internet connections
(we experienced this!)

# A simpler solution: object storage

Only two operations:

- Put an object at a key
- Retrieve an object from its key

(and a few others)

Sufficient for many applications!

# A simpler solution: object storage

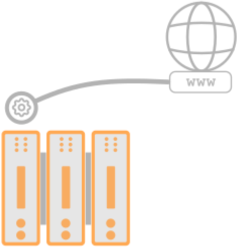
S3: a de-facto standard, many compatible applications

MinIO is self-hostable but not suited for geo-distributed deployments

# But what is Garage, exactly?

**Garage is a self-hosted drop-in replacement for the Amazon S3 object store**

that implements resilience through geographical redundancy on commodity hardware



Host a website

Store Media

Backup Target

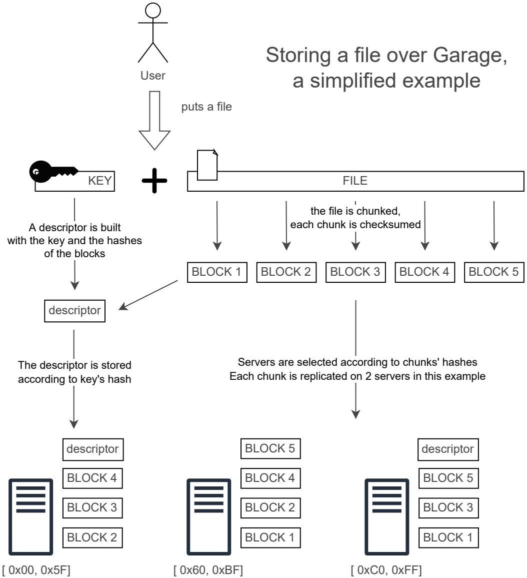
# What makes Garage different?

**Coordination-free:**

- No Raft or Paxos
- Internal data types are CRDTs
- All nodes are equivalent (no master/leader/index node)

→ less sensitive to higher latencies between nodes

# Storing and retrieving files



Storing a file over Garage, a simplified example

User

puts a file

KEY

FILE

A descriptor is built with the key and the hashes of the blocks

the file is chunked, each chunk is checksumed

BLOCK 1

BLOCK 2

BLOCK 3

BLOCK 4

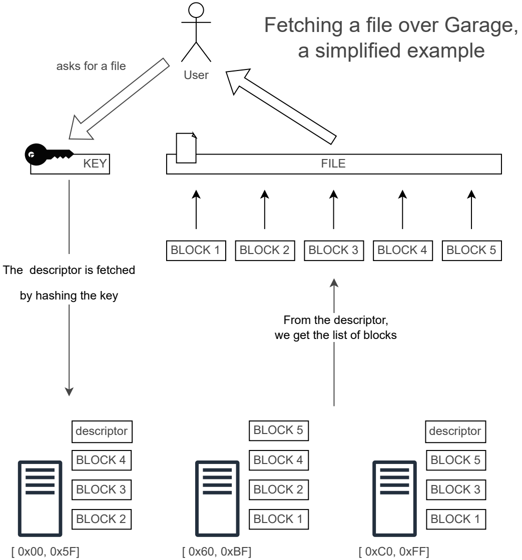
BLOCK 5

descriptor

The descriptor is stored according to key's hash

Servers are selected according to chunks' hashes
Each chunk is replicated on 2 servers in this example

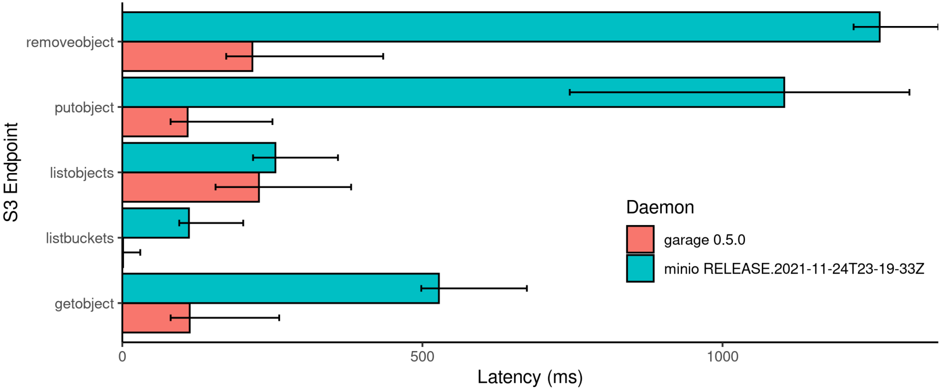
# Storing and retrieving files



# What makes Garage different?

S3 endpoint latency in a simulated geo-distributed cluster

100 measurements, 6 nodes in 3 DC (2 nodes/DC), 100ms RTT + 20ms jitter between DC
no contention: latency is due to intra-cluster communications
colored bar = mean latency, error bar = min and max latency



Get the code to reproduce this graph at https://git.deuxfleurs.fr/quentin/benchmarks

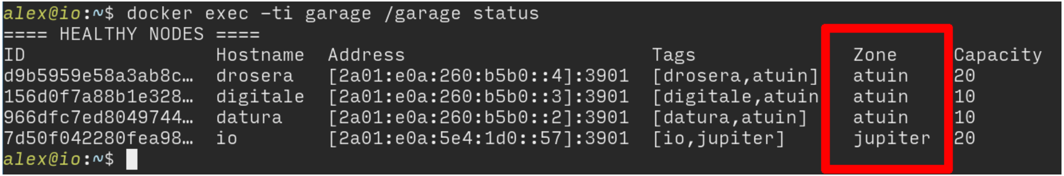
# What makes Garage different?

**Consistency model:**

- Not ACID (not required by S3 spec) / not linearizable
- **Read-after-write consistency**
  (stronger than eventual consistency)

# What makes Garage different?

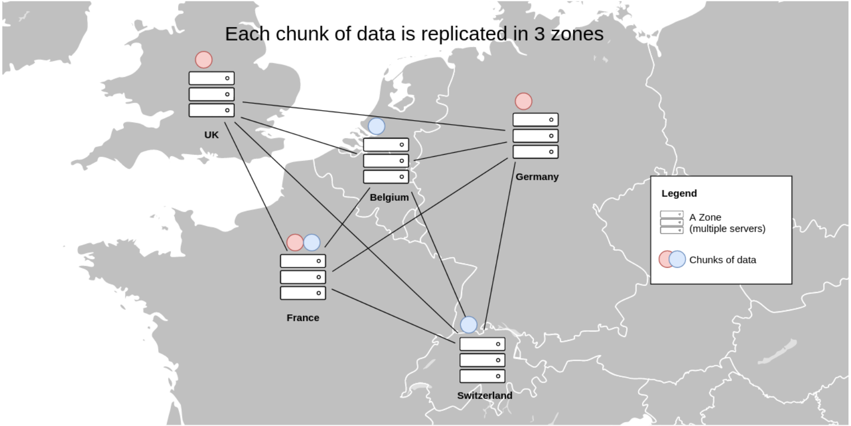
**Location-aware:**



```
alex@io:~$ docker exec -ti garage /garage status
==== HEALTHY NODES ====
ID                 Hostname  Address                      Tags              Zone     Capacity
d9b5959e58a3ab8c…  drosera   [2a01:e0a:260:b5b0::4]:3901  [drosera,atuin]   atuin    20
156d0f7a88b1e328…  digitale  [2a01:e0a:260:b5b0::3]:3901  [digitale,atuin   atuin    10
966dfc7ed8049744…  datura    [2a01:e0a:260:b5b0::2]:3901  [datura,atuin]    atuin    10
7d50f042280fea98…  io        [2a01:e0a:5e4:1d0::57]:3901  [io,jupiter]      jupiter  20
alex@io:~$
```

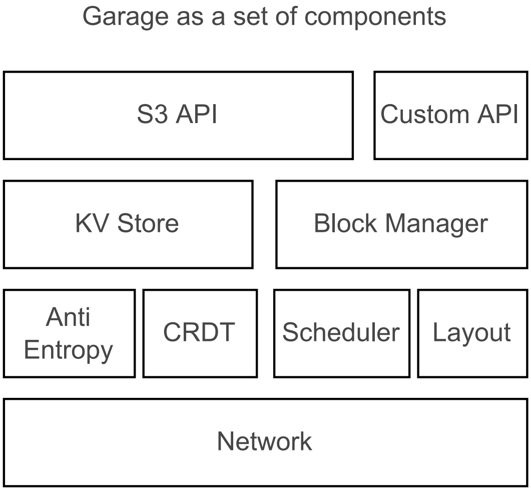
Garage replicates data on different zones when possible

# What makes Garage different?



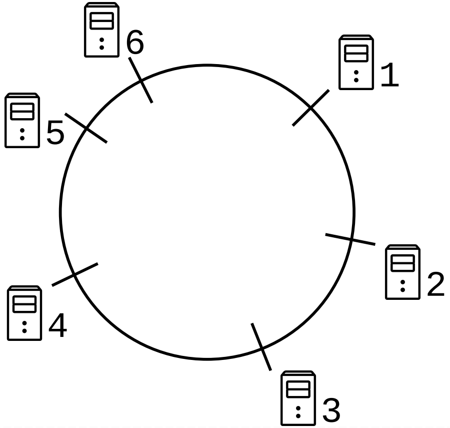
Each chunk of data is replicated in 3 zones

# Garage's architecture



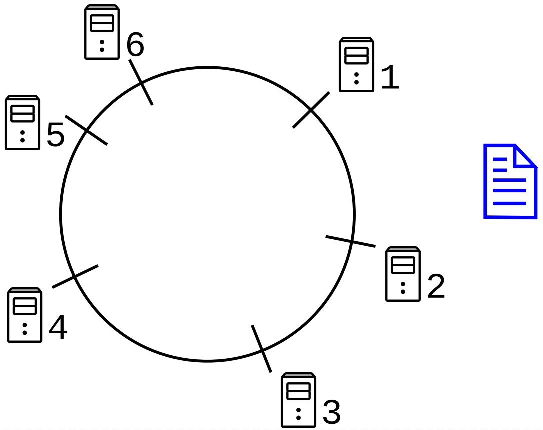
# Consistent Hashing (DynamoDB)

**How to spred files over different cluster nodes?**



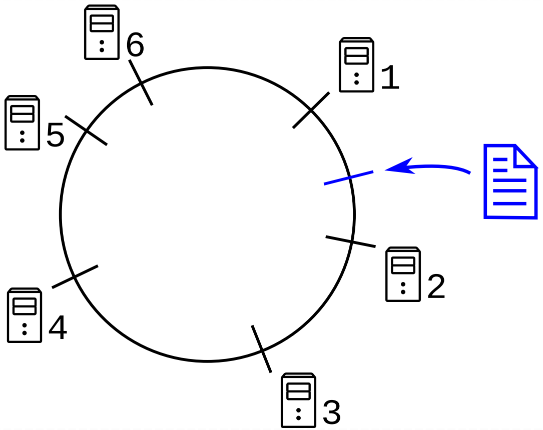
# Consistent Hashing (DynamoDB)

**How to spred files over different cluster nodes?**



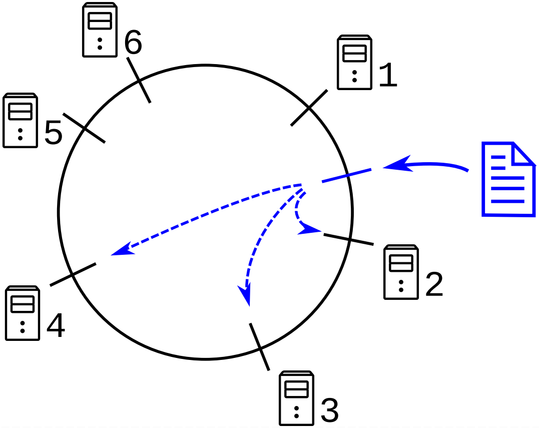
# Consistent Hashing (DynamoDB)

**How to spred files over different cluster nodes?**

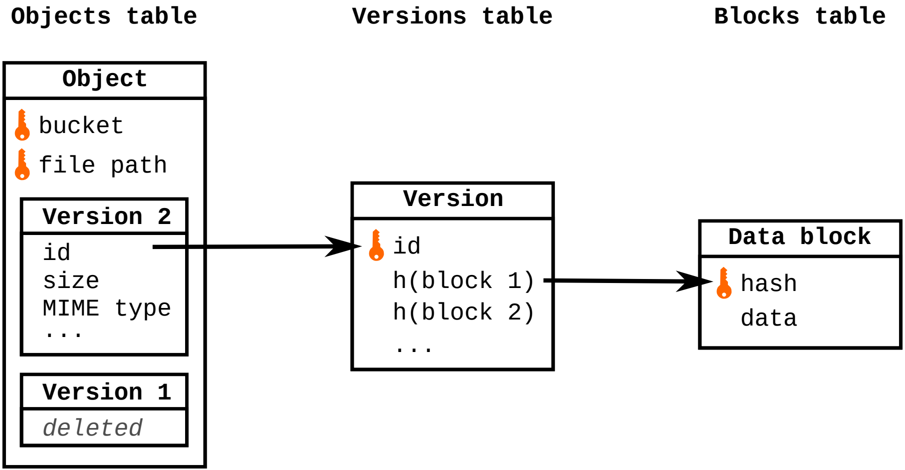


# Consistent Hashing (DynamoDB)

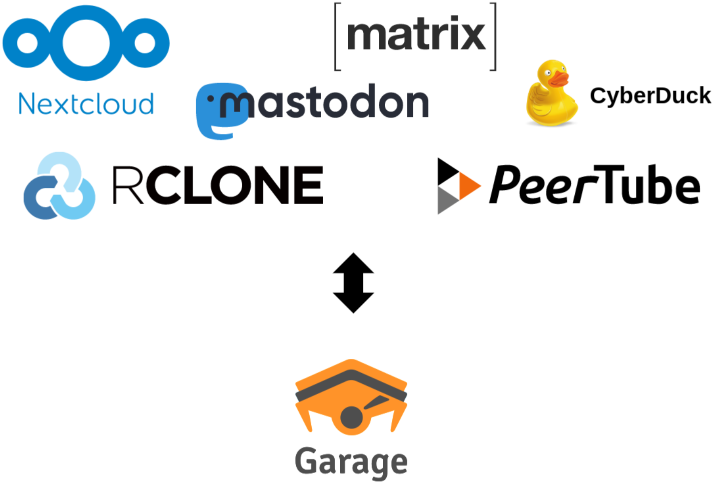
**How to spred files over different cluster nodes?**



# Garage data structures: 3 levels of consistent hashing



# An ever-increasing compatibility list



# Get Garage now!

Garage

`https://garagehq.deuxfleurs.fr/`

Matrix channel: `#garage:deuxfleurs.fr`

# Demo time!