

# AEROGRAMME

A multi-region IMAP server

FOSDEM 2024

Brussels, Belgium

*By Internet Mail*

# Some context

## About me

Quentin Dufour, Freelance developer  
PhD in distributed systems  
quentin@dufour.io

## About the Deuxfleurs collective

Non-profit collective member of CHATONS.org  
Building a small appropriated low-tech Internet

## About Aerogramme

Started in 2022, a Deuxfleurs project  
Supported by NLnet

# The problem we want to solve

Why people use emails?

Making other people available  
when it would be otherwise impossible.


What does it mean on the tech side?

Systems must be available  
otherwise they are useless



# Today's talk is about 3 ideas

- (1) Cloud & hosting providers can fail, they should not be solely relied upon.
- (2) Relaxing consistency has virtues, but correctness is mandatory.
- (3) New designs in the email ecosystem are possible in the real world



**Don't trust your provider**

# Cloud/hosting providers can fail hard

Global Switch datacenter, Clichy

Interxion datacenter, La Courneuve

DATA4, Saclay

TeleHouse 2, Paris

The Register

**Google Cloud's watery Parisian outage enters third week, with no end in sight**

To make matters worse, other bits of the same region have wobbled

Simon Sharwood

Wed 10 May 2023 04:30 UTC

Google europe-west9, april 2023 incident

<https://cloud.google.com/network-connectivity/docs/interconnect/concepts/choosing-colocation-facilities>

# Moving to reliability-first designs

Gmail and Google Search reliability is built into their source code, not Google's DC.  
FLOSS should start writing reliable software too!



Cloud Native Patterns by Cornelia Davis

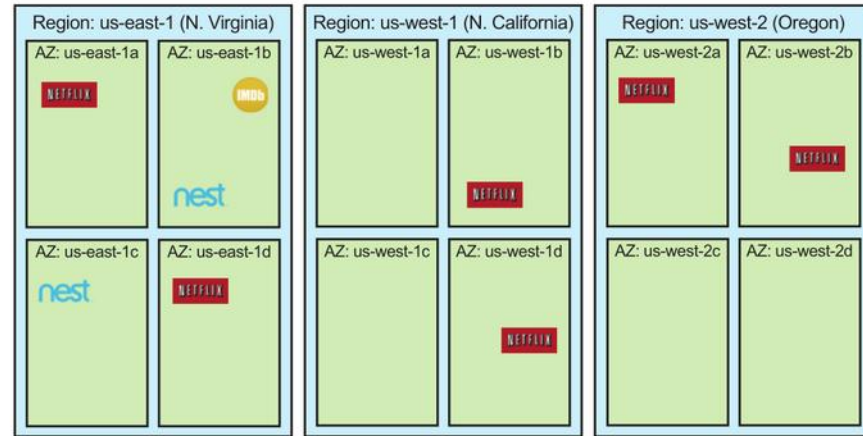


Figure 1.2 Applications deployed onto AWS may be deployed into a single AZ (IMDb), or in multiple AZs (Nest) but only a single region, or in multiple AZs and multiple regions (Netflix). This provides different resiliency profiles.

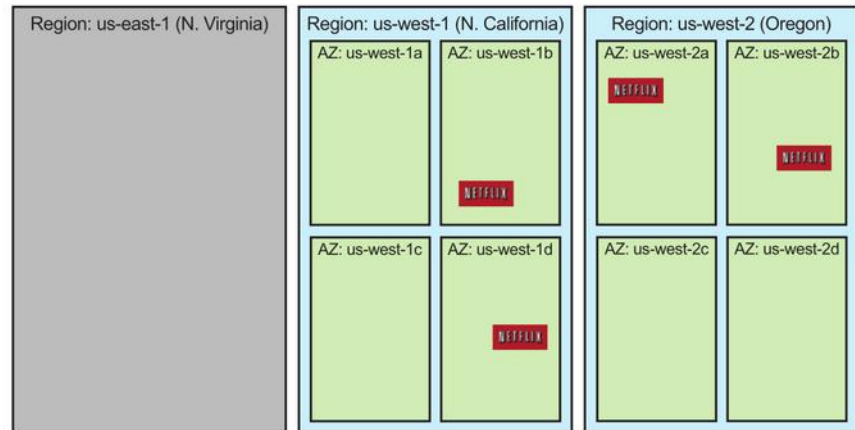
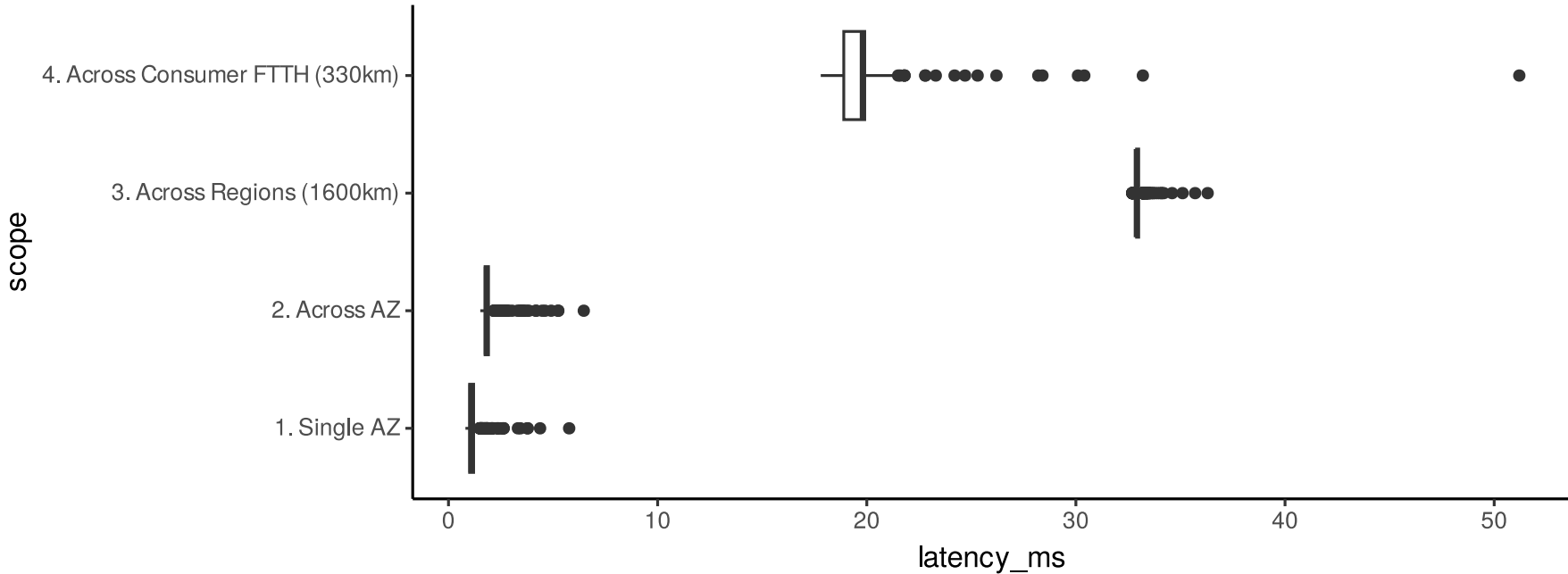


Figure 1.3 If applications are properly architected and deployed, digital solutions can survive even a broad outage, such as of an entire region.

# Reliable software are hard to write

Especially when you can't neglect latency & crashes anymore.  
It's called distributed computing/systems.



*Measurements done on Scaleway from PAR1 to PAR1(1), PAR2(2), WAR1(3).  
1k ICMP packets, 100ms interval, on 2024-01-29, using DEV1-S Ubuntu instances.*

**15×**

Delays are 15× higher in a multi-region deployment compared to a single region one.



A decorative border consisting of alternating red and blue diagonal stripes surrounds the central text. The stripes are parallel and oriented at a 45-degree angle.

**Relaxing consistency  
while staying correct**

# Apache James summarizes the problem

Note : Quote reworded for the sake of fitting the slide.

*Scaling emails infrastructure is a notoriously hard problem as we rely on **monotonic UID generation**.*

*Running the Distributed Server IMAP server in a multi datacenter setup without strong consistency will likely result in data loss as the same UIDs could be allocated several times. With strong consistency, it will result in very slow operations*

*Running James with a multi data-center Cassandra setup is discouraged.*

<https://james.staged.apache.org/james-project/3.6.0/servers/distributed/architecture/consistency-model.html>

# Review of existing high-availability approaches

## Leader/follower designs

Cyrus IMAP, Dovecot

→ No high availability

## Consensus/Total Order based designs

Stalwart IMAP/JMAP

Apache James

Wildduck

→ No multi-region, latency sensitive

## CRDT designs

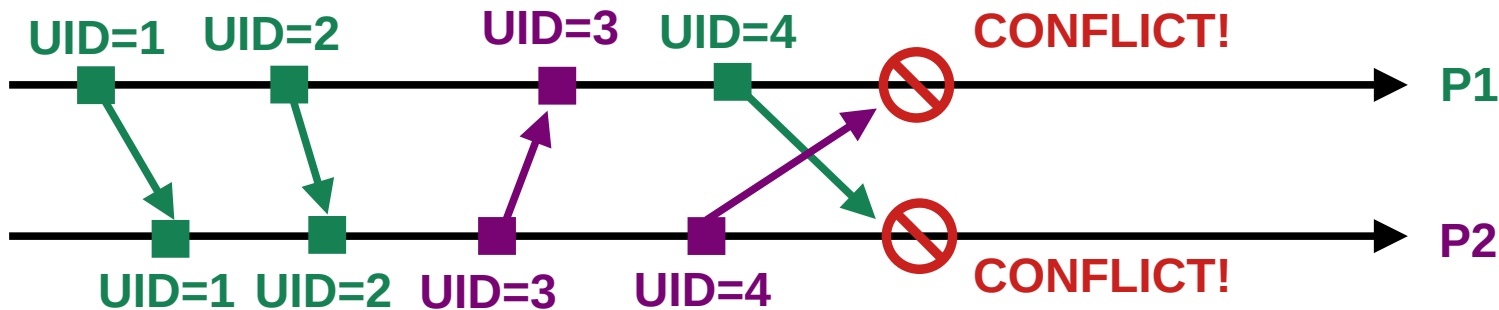
Pluto

→ Incomplete implementation, missing UID

# Our solution: living with conflicts

Conflicts are OK in IMAP as long as 1) they are detected and 2) UIDVALIDITY is changed. Downside: It will trigger a full, expensive resynchronization for the clients.

## How UID conflicts happen?



## Our implementation

- 1** Event log is not totally ordered but causally ordered
- 2** Proven algorithm to solve conflicts and compute a new UIDVALIDITY
- 3** Clever sync of the event log to reduce the conflict window

Proof: <https://aerogramme.deuxfleurs.fr/documentation/internals/imap-uid/>

# "But you are cheating!"

"You did not solve the problem of monotonic UID, you changed the problem!  
And it's not without impact on the end-user!"

Better than (wrongly) tweaking Raft

Kubernetes stale reads [1]


Github Orchestrator SQL corruption [2]

Optimist approaches are now safe

eg. simple frontend multiplexer

[1]: <https://github.com/kubernetes/kubernetes/issues/59848>

[2]: <https://github.blog/2018-10-30-oct21-post-incident-analysis/>



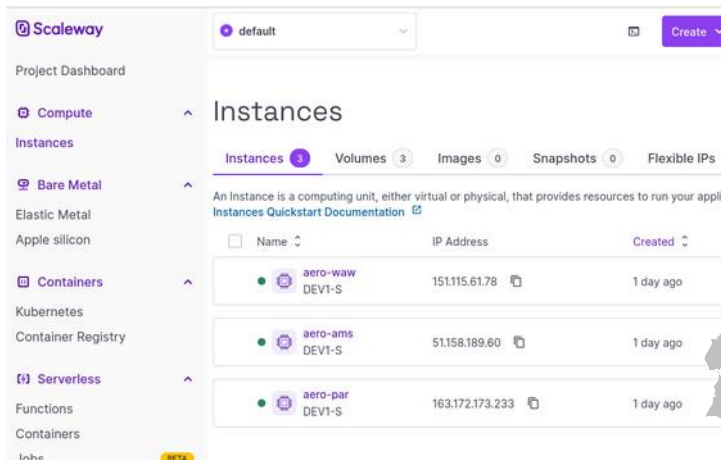
**Talk is cheap,  
show me the mail server!**

# A multi-region deployment

```
$ dig +short MX saint-ex.deuxfleurs.org
10 aero-ams.machine.deuxfleurs.org.
10 aero-par.machine.deuxfleurs.org.
10 aero-war.machine.deuxfleurs.org.
```

```
$ dig +short imap.saint-ex.deuxfleurs.org
saint-ex.deuxfleurs.org.
51.158.189.60
151.115.61.78
163.172.173.233
```

```
$ dig +short smtp.saint-ex.deuxfleurs.org
saint-ex.deuxfleurs.org.
51.158.189.60
151.115.61.78
163.172.173.233
```

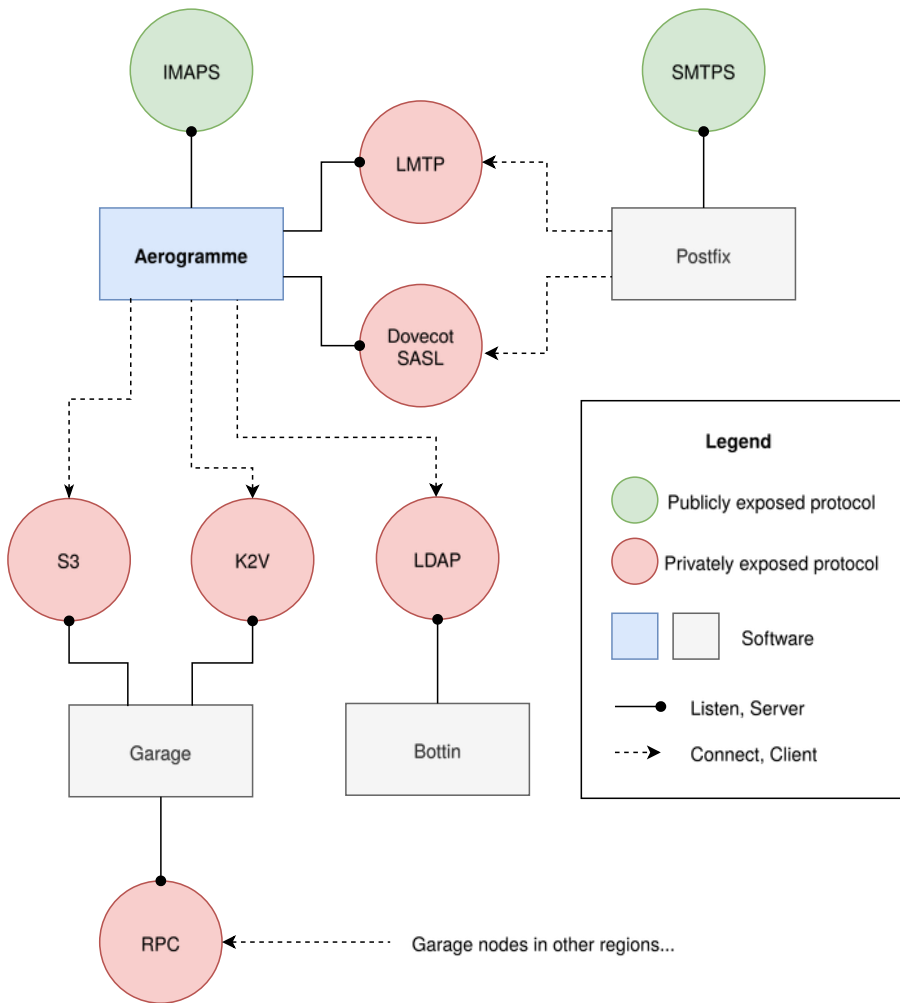


The screenshot shows the Scaleway Project Dashboard with the 'Instances' section expanded. It displays a table of three instances, each in a different region: WAW (Warsaw), AMS (Amsterdam), and PAR (Paris). Each instance is named 'aero-*region*-*machine*' and is of type 'DEVI-S'. The IP addresses and creation times are also visible.

Name	IP Address	Created
aero-waw DEVI-S	151115.61.78	1 day ago
aero-ams DEVI-S	51.158.189.60	1 day ago
aero-par DEVI-S	163.172.173.233	1 day ago



# Focusing on one region



```
root@aero-ams:~/saint-ex# docker compose up -d  
[+] Running 5/0
```

- ✓ Container saint-ex-postfix-1 Running
- ✓ Container saint-ex-garage-1 Running
- ✓ Container saint-ex-aerogramme-1 Running
- ✓ Container saint-ex-bottin-1 Running
- ✓ Container saint-ex-consul-1 Running

## Notes

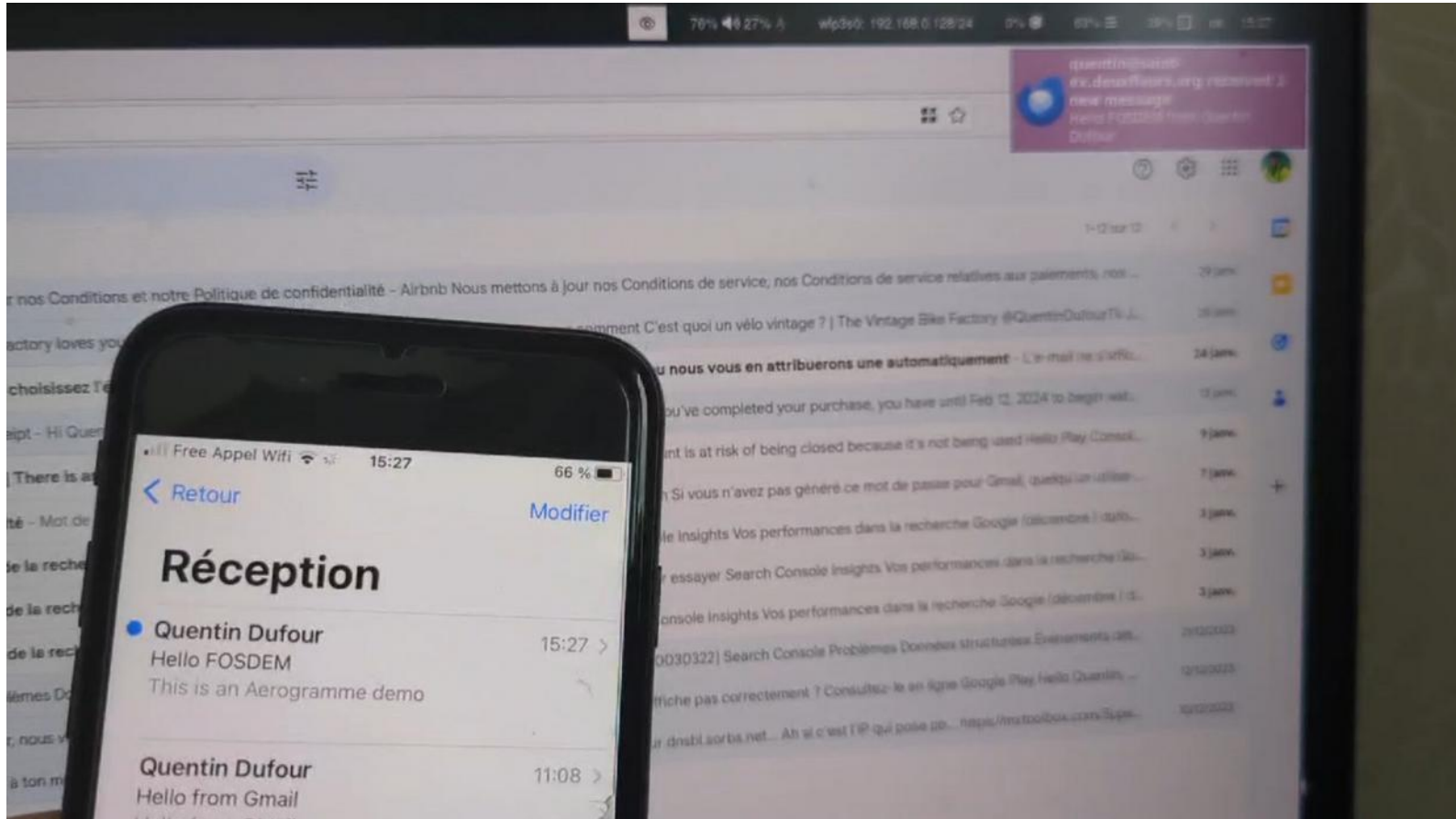
Postfix delivers emails to the local Aerogramme instance only

Each device has a session on a single random instance.

IMAP sessions = watching K2V range.  
Receiving an email = range changed.



# It seems it works...



<https://quentin.dufour.io/aerogramme-demo.mp4>

# Conclusion

## Takeaways

---

- 1) Aerogramme is designed from the ground-up for reliability
- 2) Aerogramme tolerates UID conflicts, correctly handles them, and minimizes them.
- 3) Aerogramme already works in real environments

## Future works

---

- 1) CalDAV and CardDAV
- 2) Additional user testing
- 3) Performance measurements/improvements