

Run your database like a CDN

Optimizing application performance for global users

Presented by Ben Darnell, CTO



Agenda

1. Background
2. The Journey of Movr
3. Replication, within and across regions
4. Conclusion

About CockroachDB

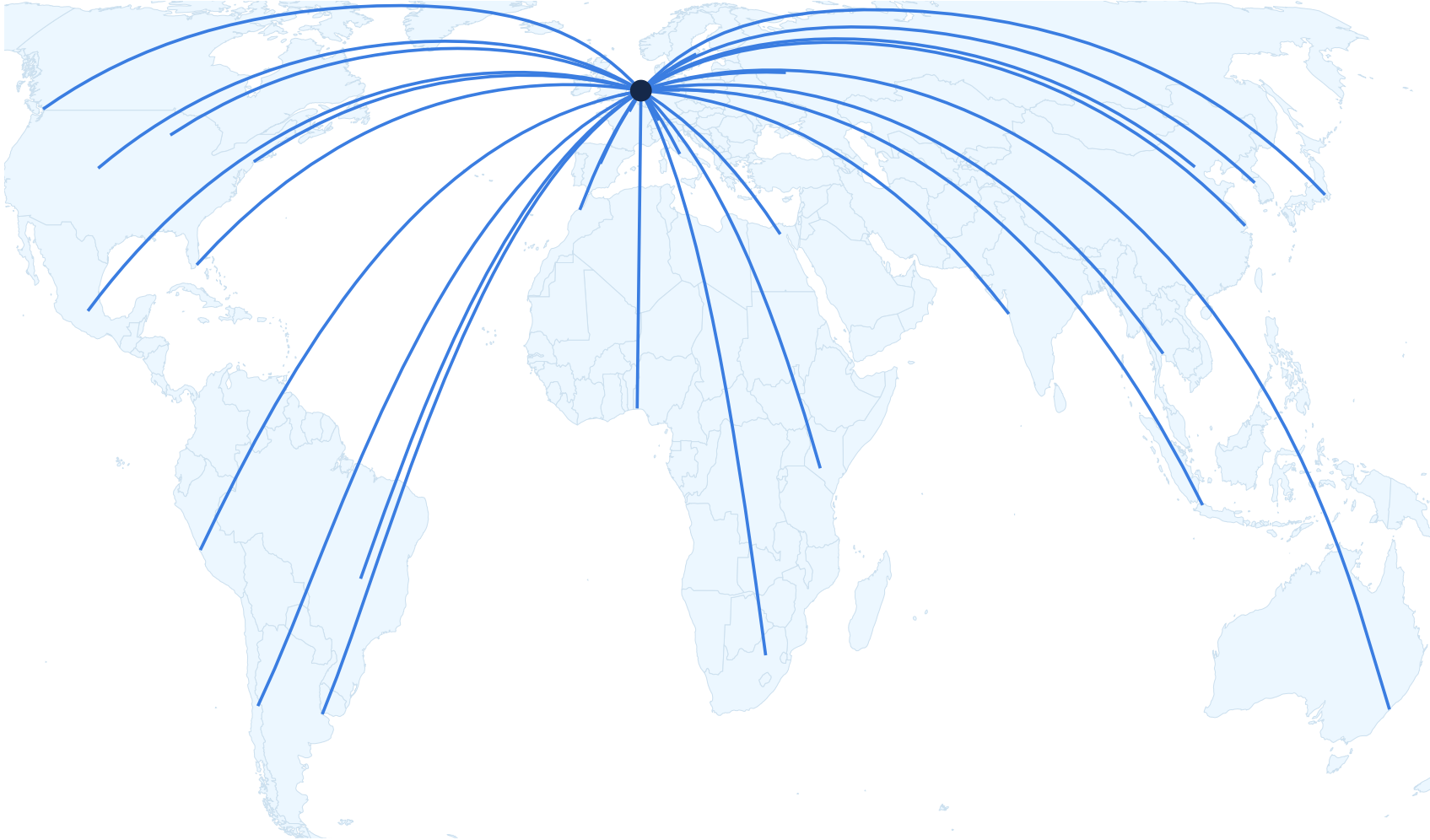
- Distributed
- Consistent
- SQL
- For transactional (OLTP) workloads

What is a CDN?

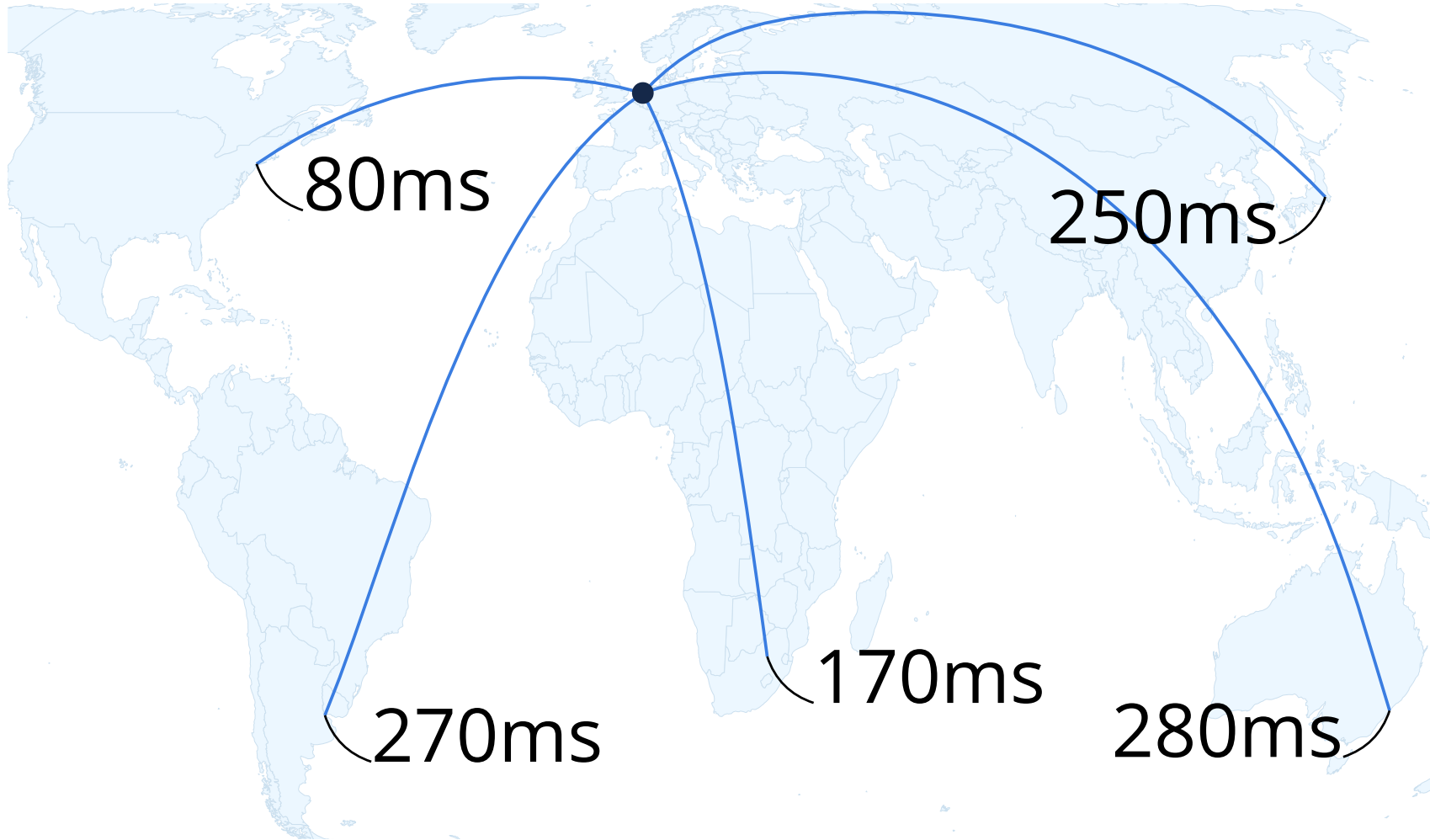
What is a CDN?

- A **Content Delivery Network** is a global network of servers and caches.
- Commonly used for static content.
- Data comes from the server closest to the user.

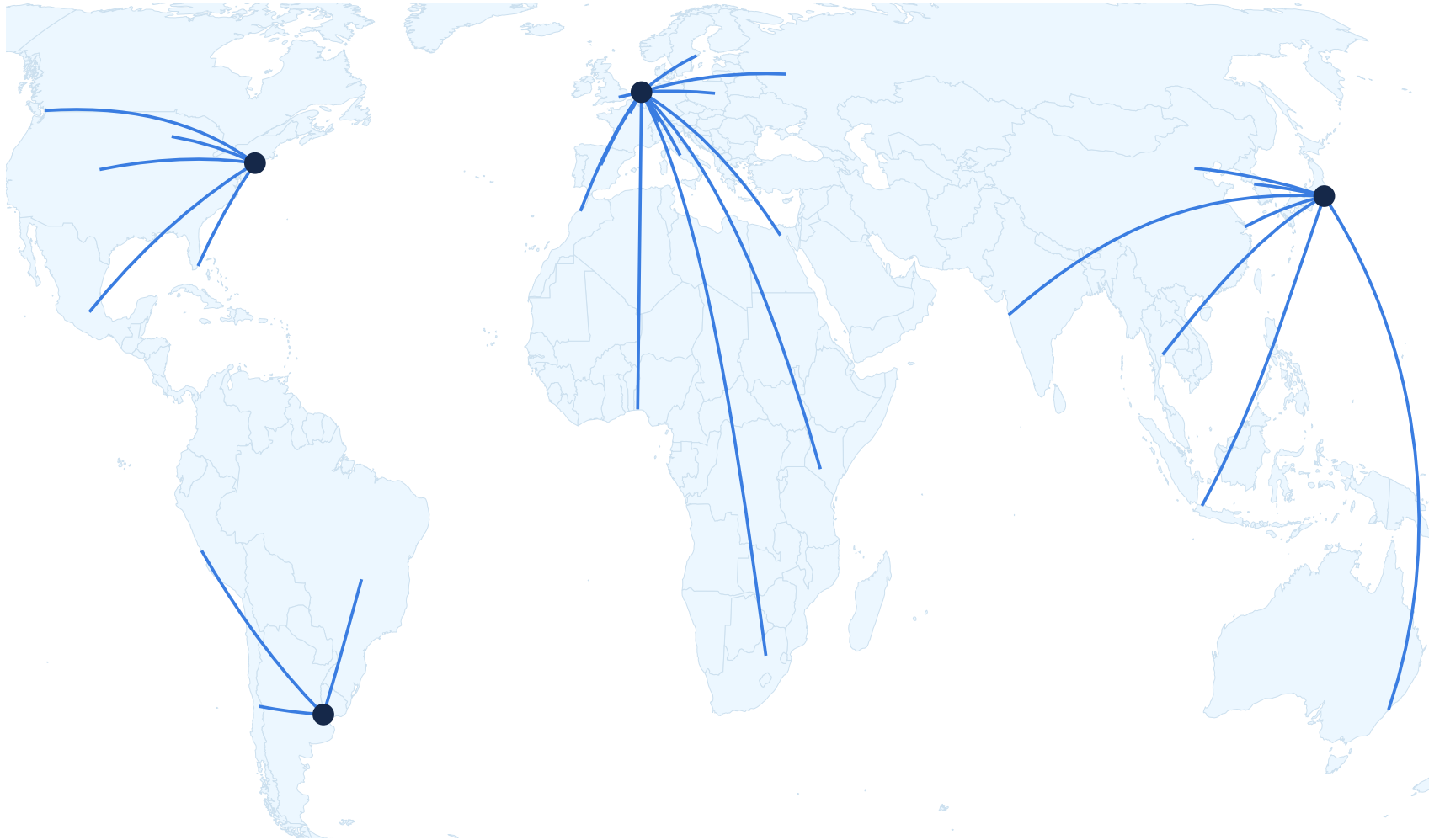
Without a CDN, all traffic goes to one place



The speed of light



With a CDN, servers are distributed around the world



CDNs improve latency for static content

Why not the same for databases?

CDN updates

- Relatively infrequent
- Mostly centralized
- Relaxed consistency

Database updates

- Very frequent
- Often customer-initiated
- Application expects consistency

Evolution of distributed data

- 1990s: Replicate for fault-tolerance
- 2000s: Shard for scalability

Evolution of distributed data

- 1990s: Replicate for fault-tolerance
- 2000s: Shard for scalability
- 2018: Distribute for performance

A database is not just a place to store data, it is a tool to get that data where it is needed.

Agenda

1. Background
2. The Journey of Movr
3. Replication, within and across regions
4. Conclusion

About Movr

- Movr is a **fictional** vehicle-sharing startup



- Launched in Amsterdam and expanding globally

Movr's Data

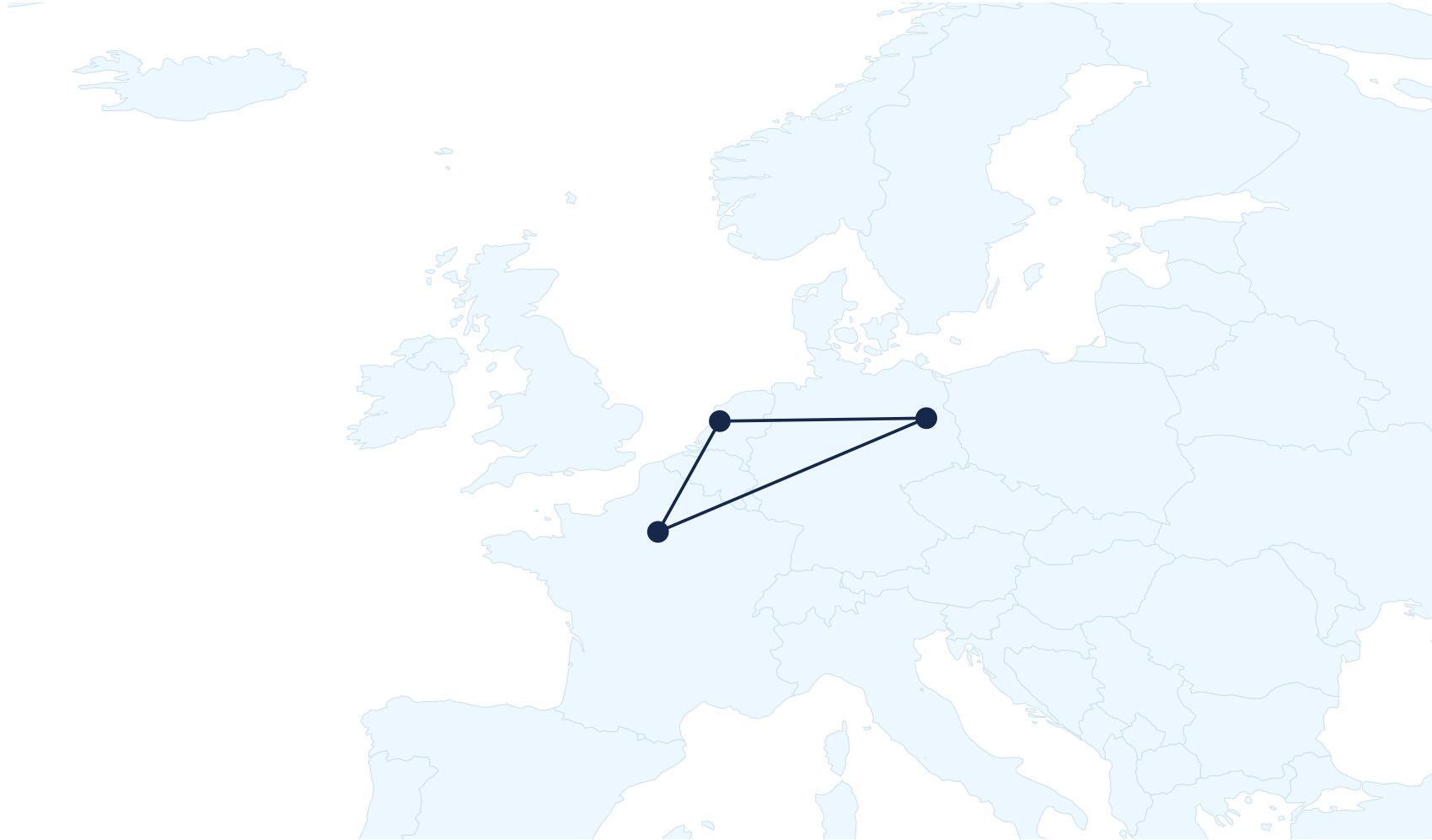
- Vehicle data
 - Tied to a city
 - Frequent updates
 - Read performance critical
- User data
 - Often, but not always, in home city
 - Cached in app

Phase 1: The first city

Movr launched in Amsterdam with a single datacenter



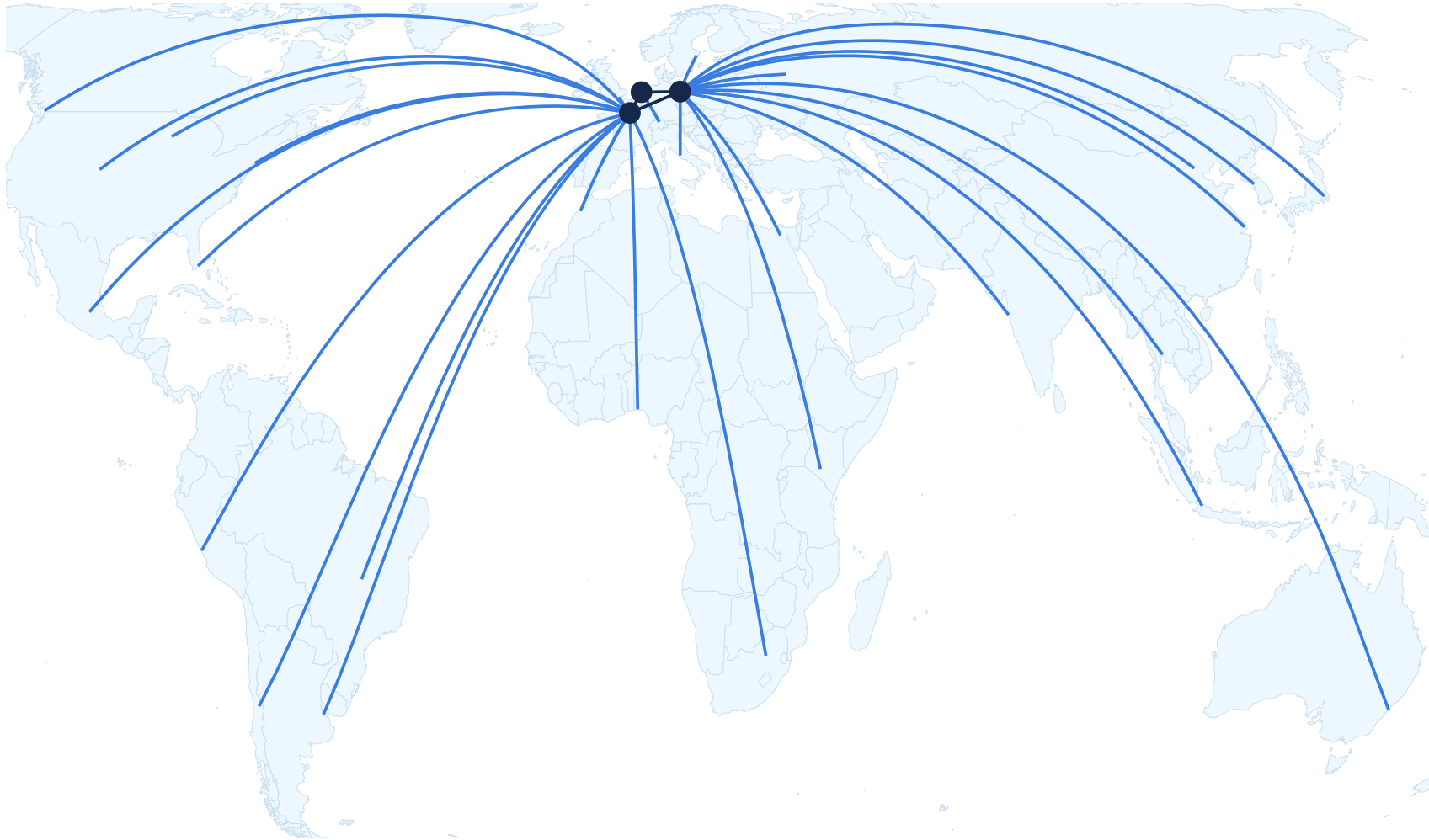
Then replicated for high availability



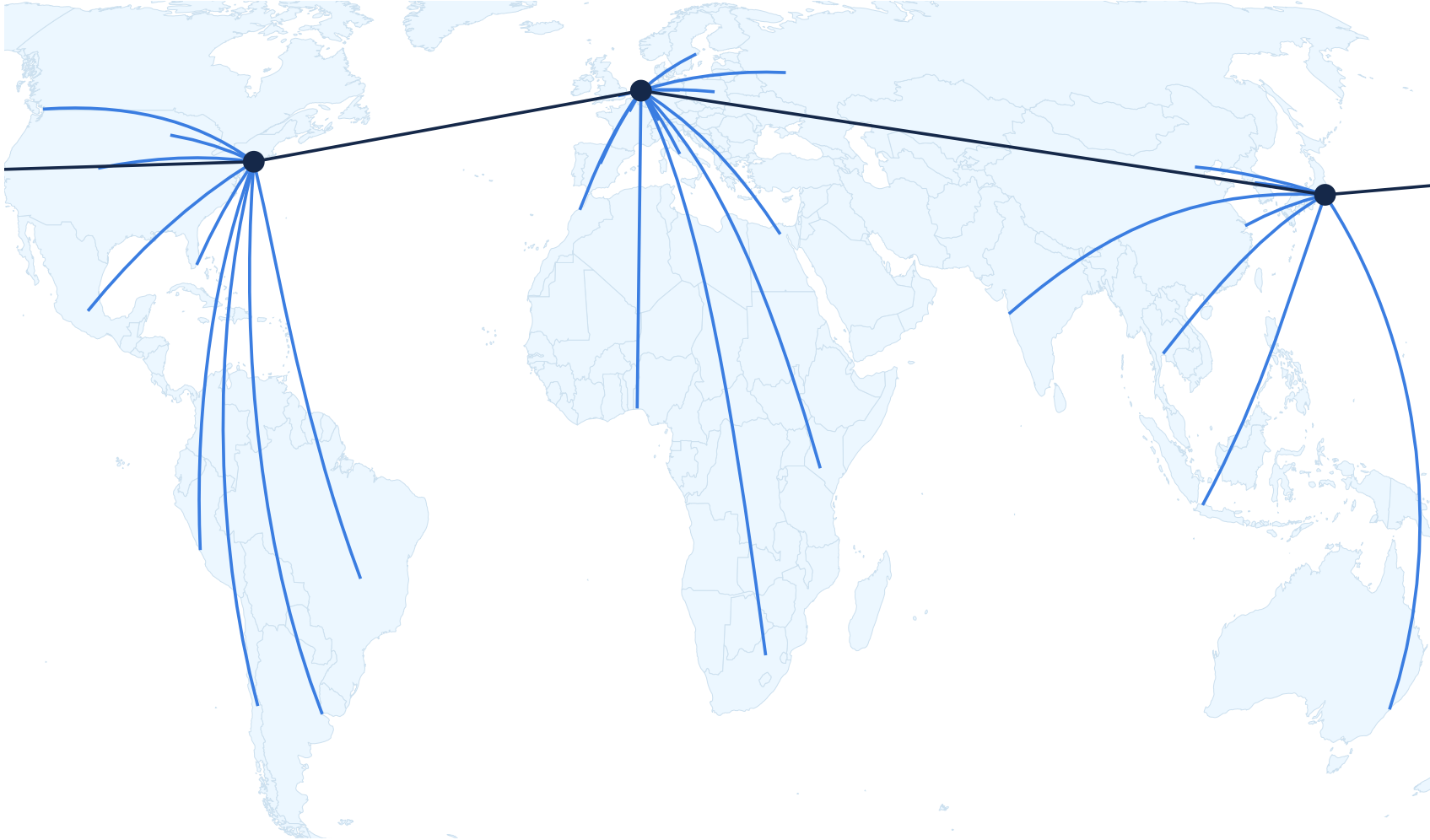
Phase 2: Regional expansion

Phase 3: The world!

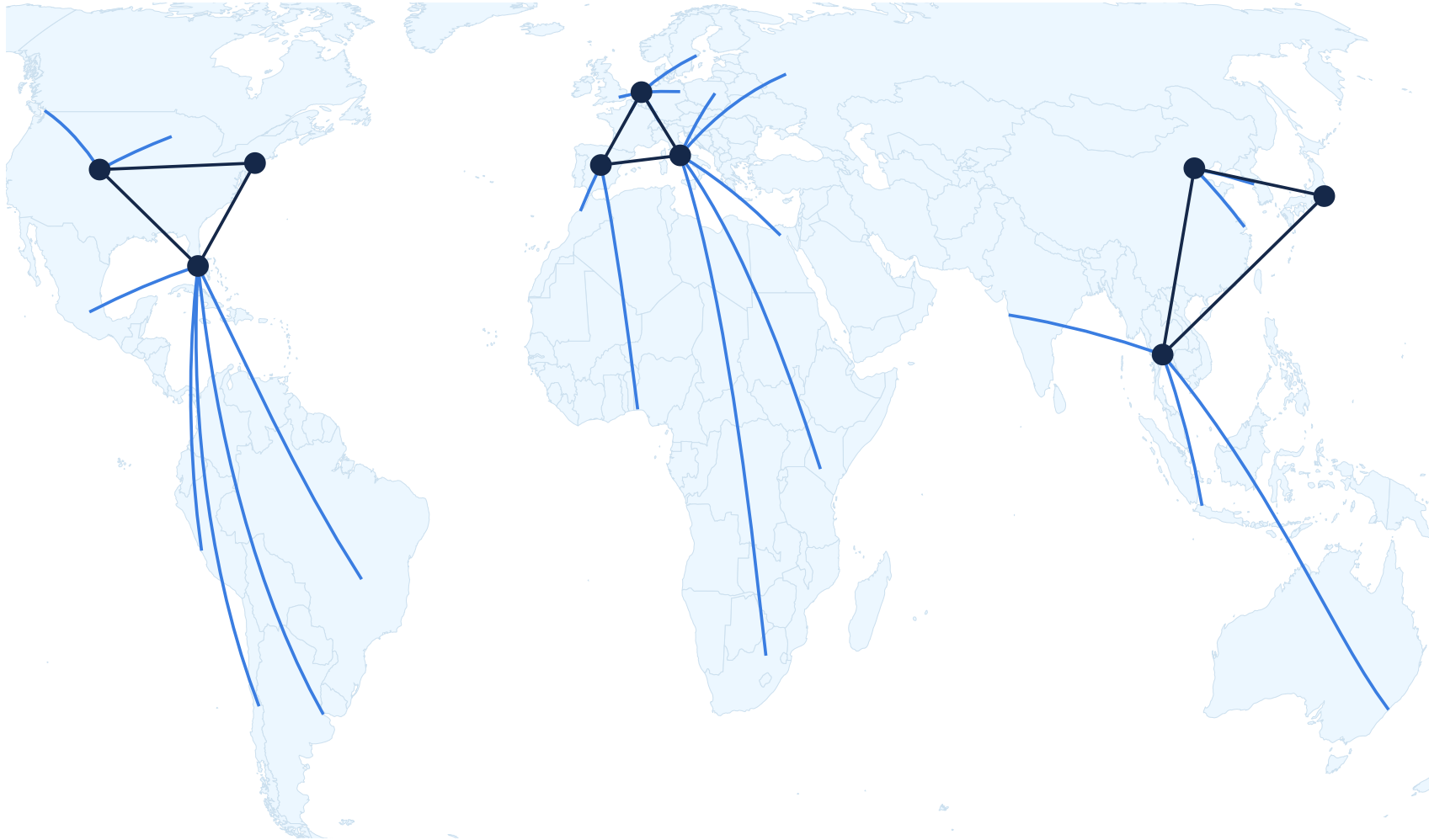
Leaving servers in Europe has poor latency



Simply distributing servers has high replication latency



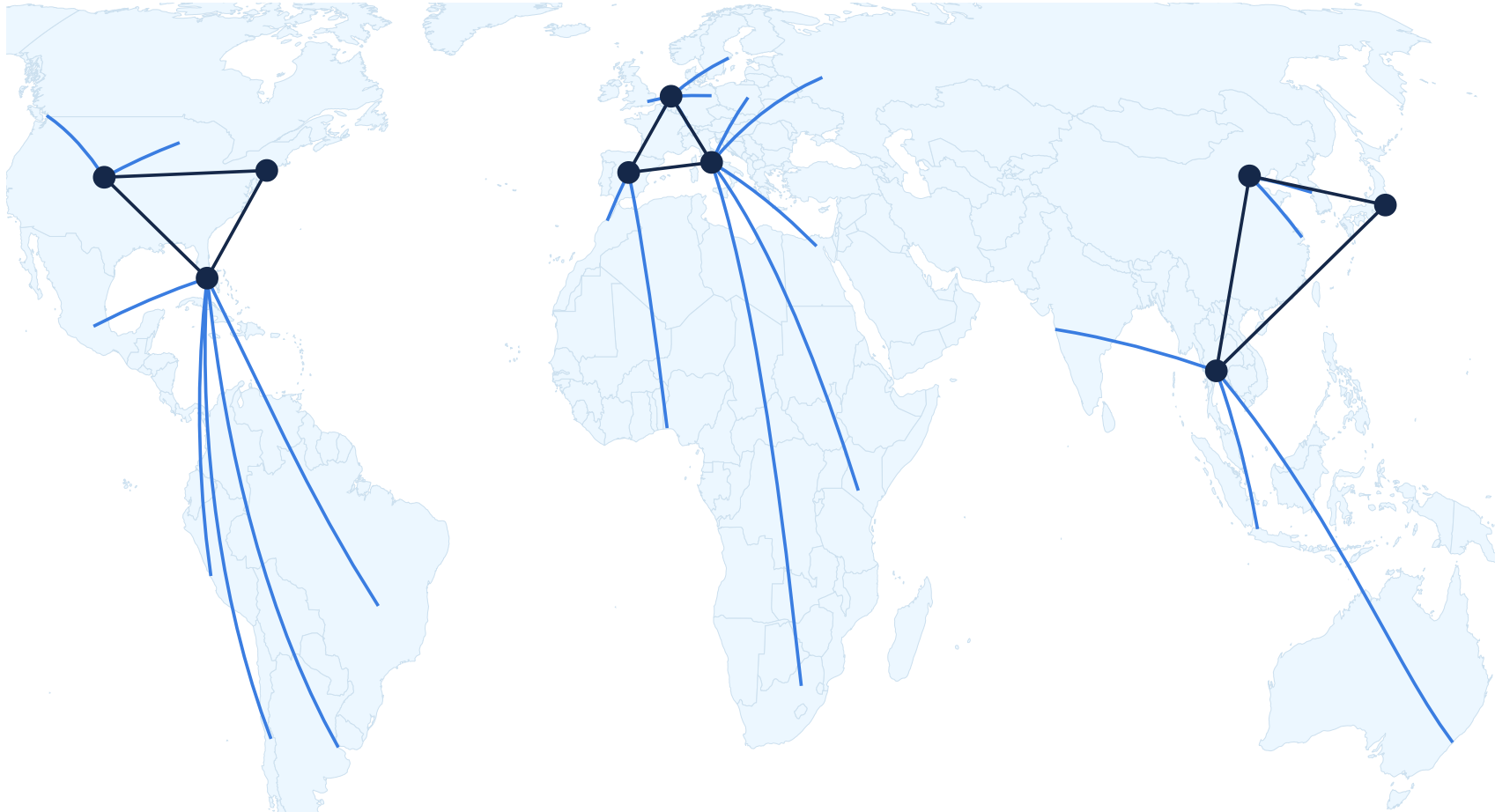
Solution: Replicate within regional sub-clusters



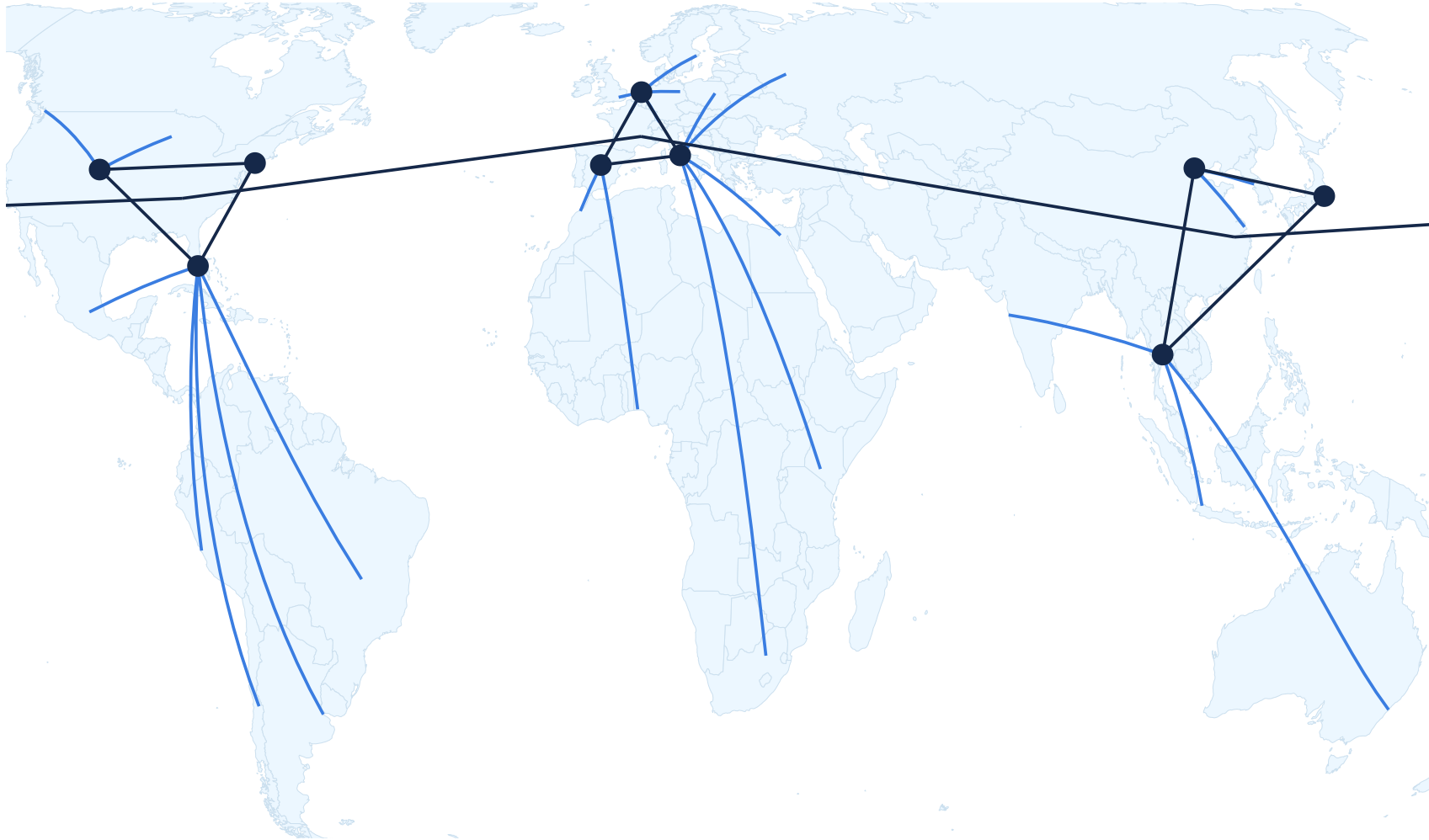
Agenda

1. Background
2. The Journey of Movr
3. Replication, within and across regions
4. Conclusion

Regional clusters could be completely separate databases...



...but in CockroachDB they can be parts of one big cluster



Replication in CockroachDB

- Looks like one logical database
- Fine-grained control over data placement
- Transactions can include data in different regions

Replication in CockroachDB

- Each record has 3+ replicas
 - One is *leader*
- Writes talk to a majority of replicas
 - Synchronous replication ensures fault tolerance and consistency
- Reads go to leader, guaranteed up to date
 - May not be nearest

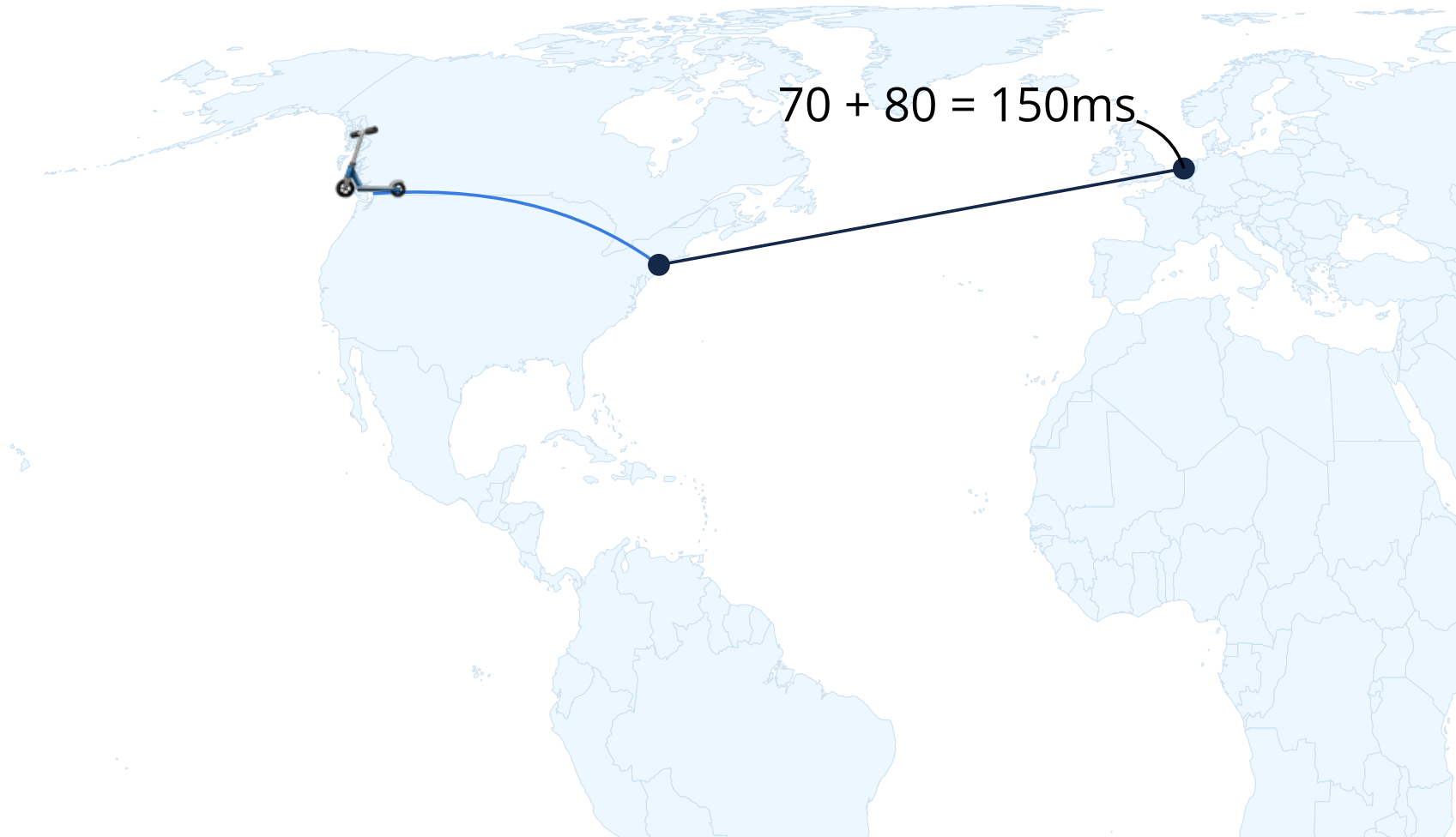
Movr's vehicle data

- Vehicle data is tied to a city, so keep it in region
- Global replication is OK for reads but makes writes slow

Reads from Vancouver may be served from New York



Writes go to New York and Amsterdam



Configuring servers

```
ams1$ cockroach start --locality=region=eu
nyc1$ cockroach start \
    --locality=region=usa --join=ams1
tok1$ cockroach start \
    --locality=region=asia --join=ams1
```

Partitioning vehicle data

```
CREATE TABLE vehicle (  
  country STRING(2),  
  id UUID,  
  attrs JSONB,  
  PRIMARY KEY (country, id))  
PARTITION BY LIST (country)  
  europe VALUES IN ('nl', 'fr', 'de'...),  
  americas VALUES IN ('us', 'ca', 'mx'...);
```


Configuring replication

```
echo 'constraints: {"+region=eu": 3}' | \  
cockroach zone set movr.vehicles.europe
```

Writes from Vancouver now go to Denver and New York



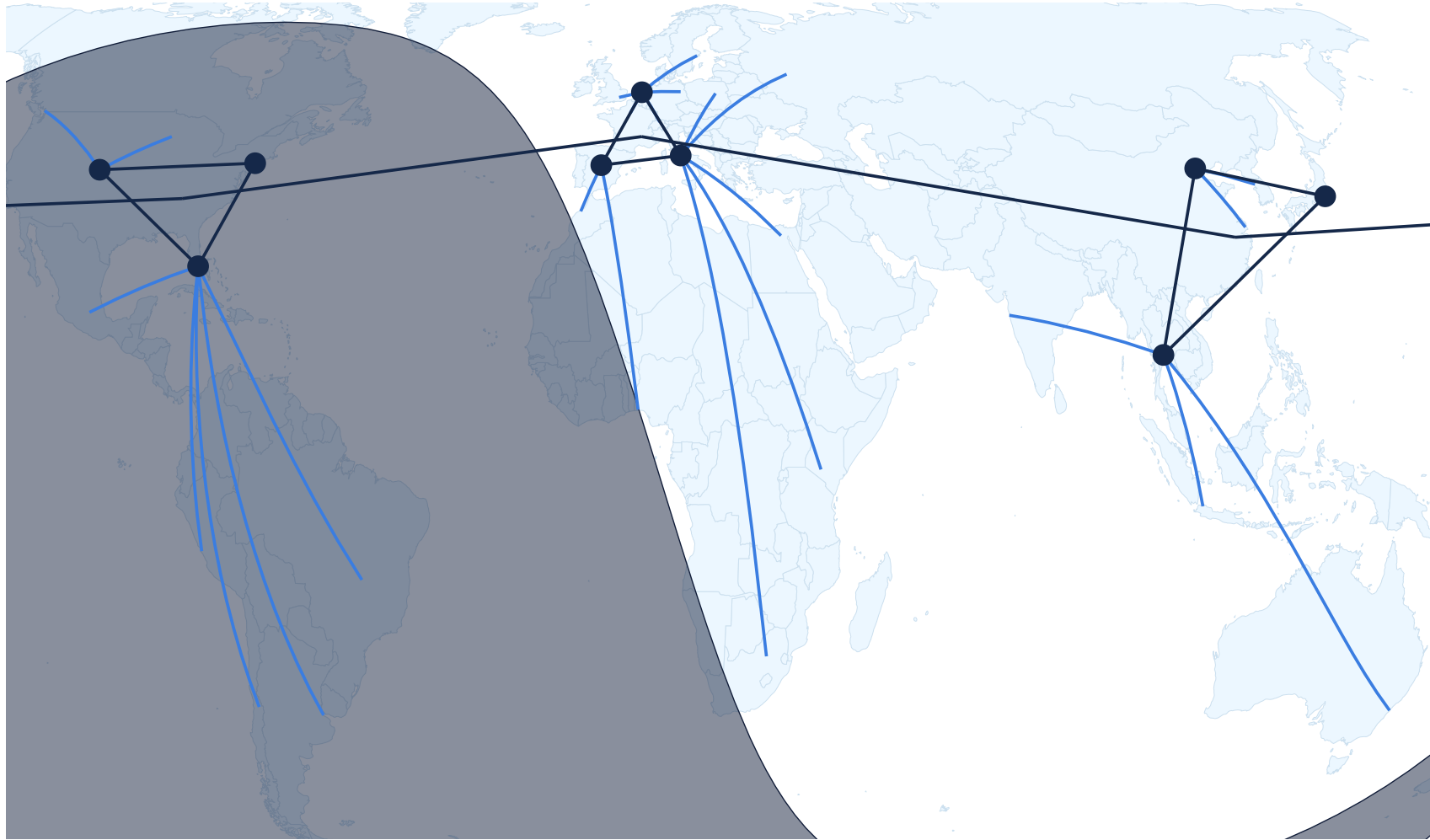
Movr's user data

- User data could be replicated in their home region or globally
 - Write performance is best in-region
 - Read performance better if global

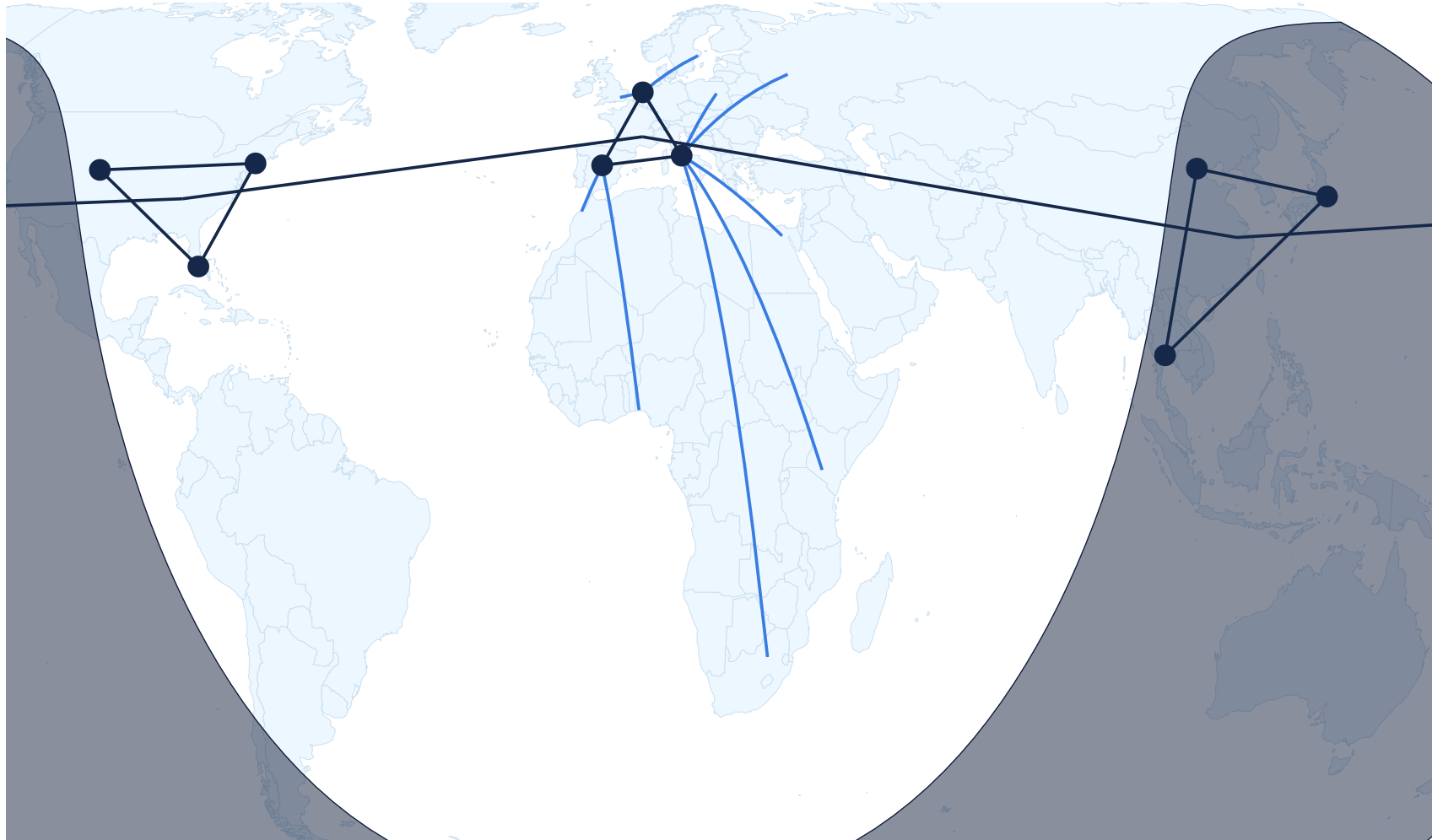
Replicate user data globally

- Writes are slow, but less frequent
- Read performance depends on where the leader is

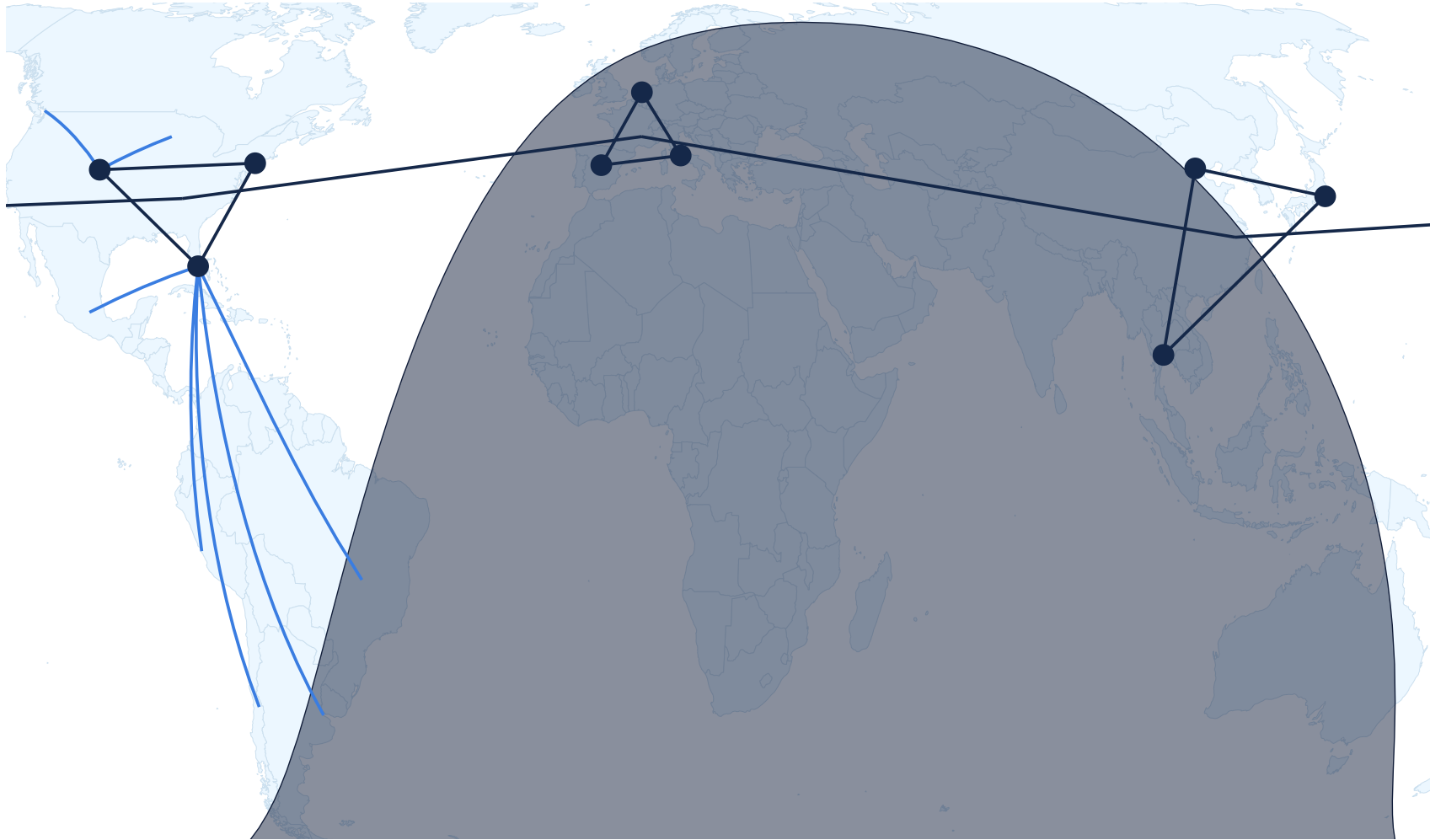
Leadership follows the sun



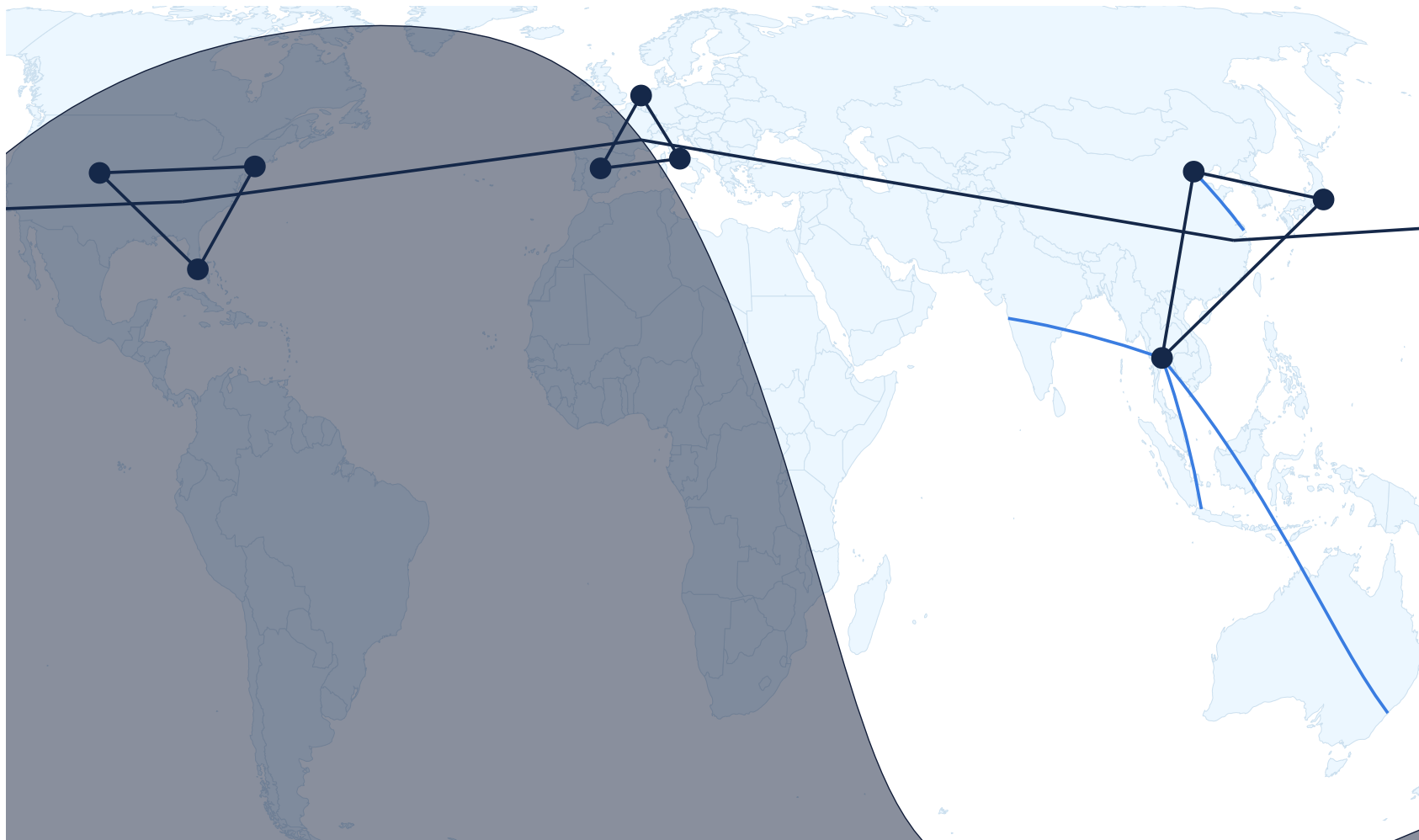
The region with the most traffic becomes leader



The region with the most traffic becomes leader



The region with the most traffic becomes leader



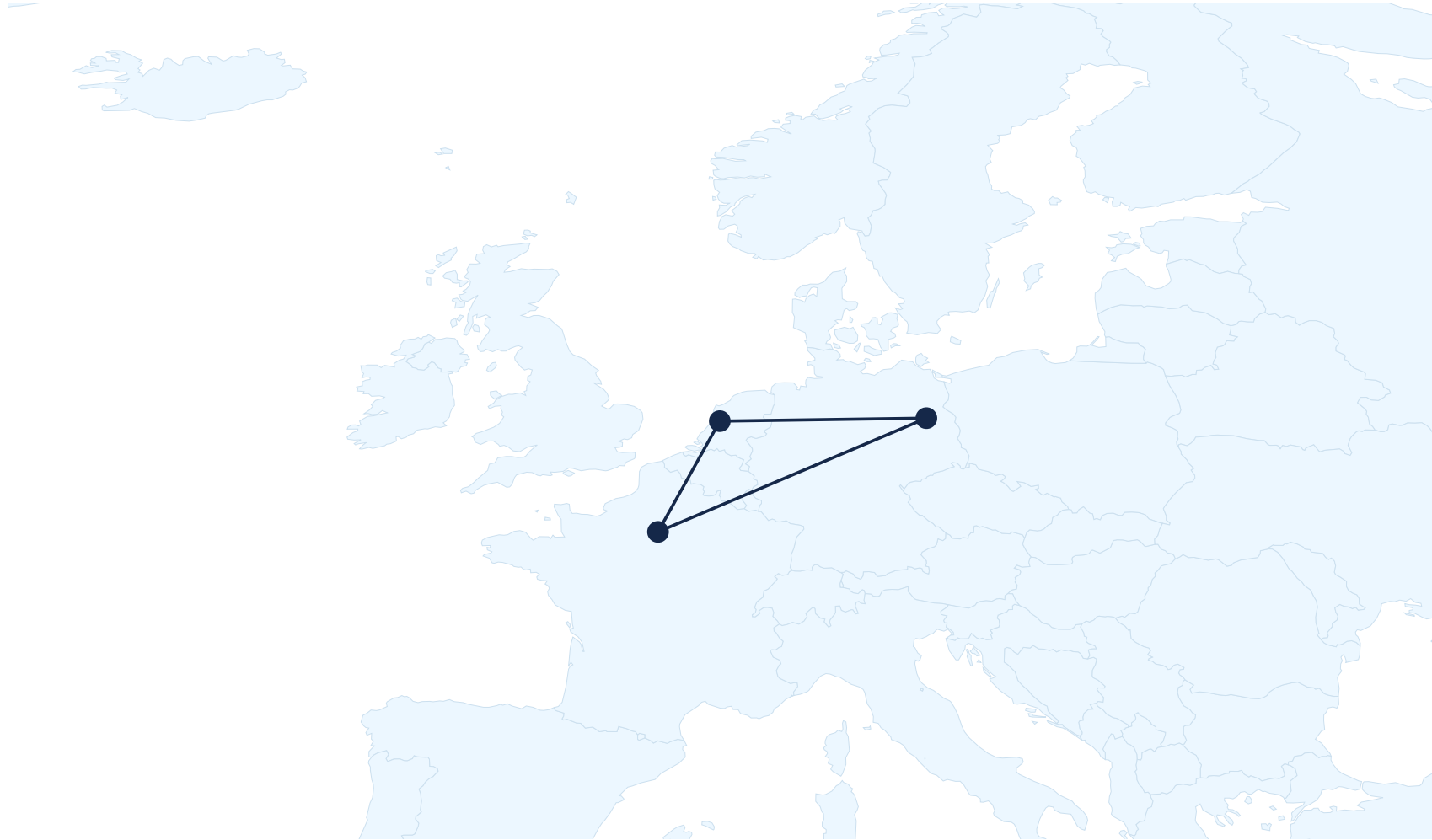
Agenda

1. Background
2. The Journey of Movr
3. Replication, within and across regions
4. Conclusion

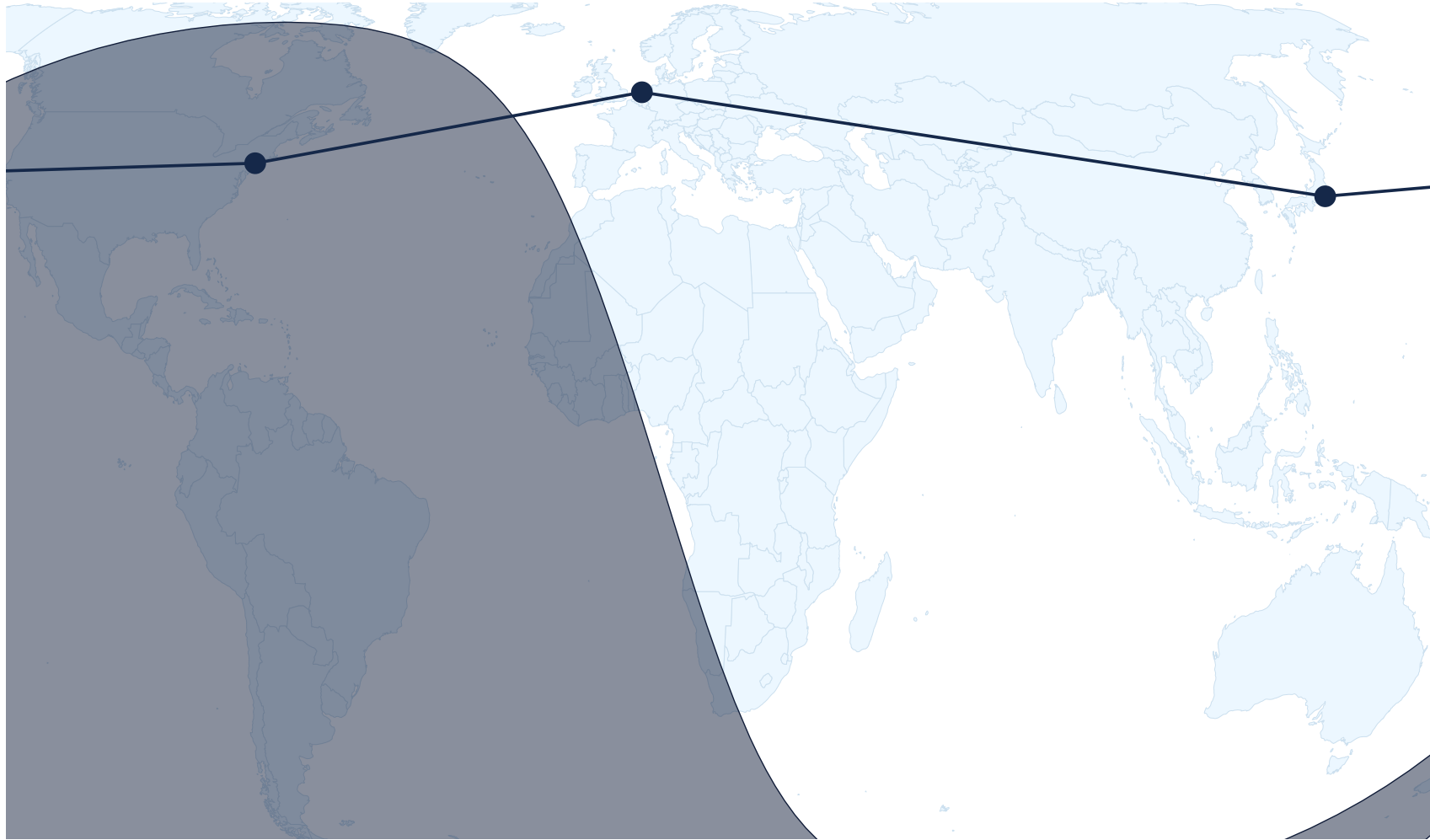
Conclusion

- Your users are global, your data should be too
- Think of geographic replication as a tool to improve performance
- Different tables may need different replication strategies

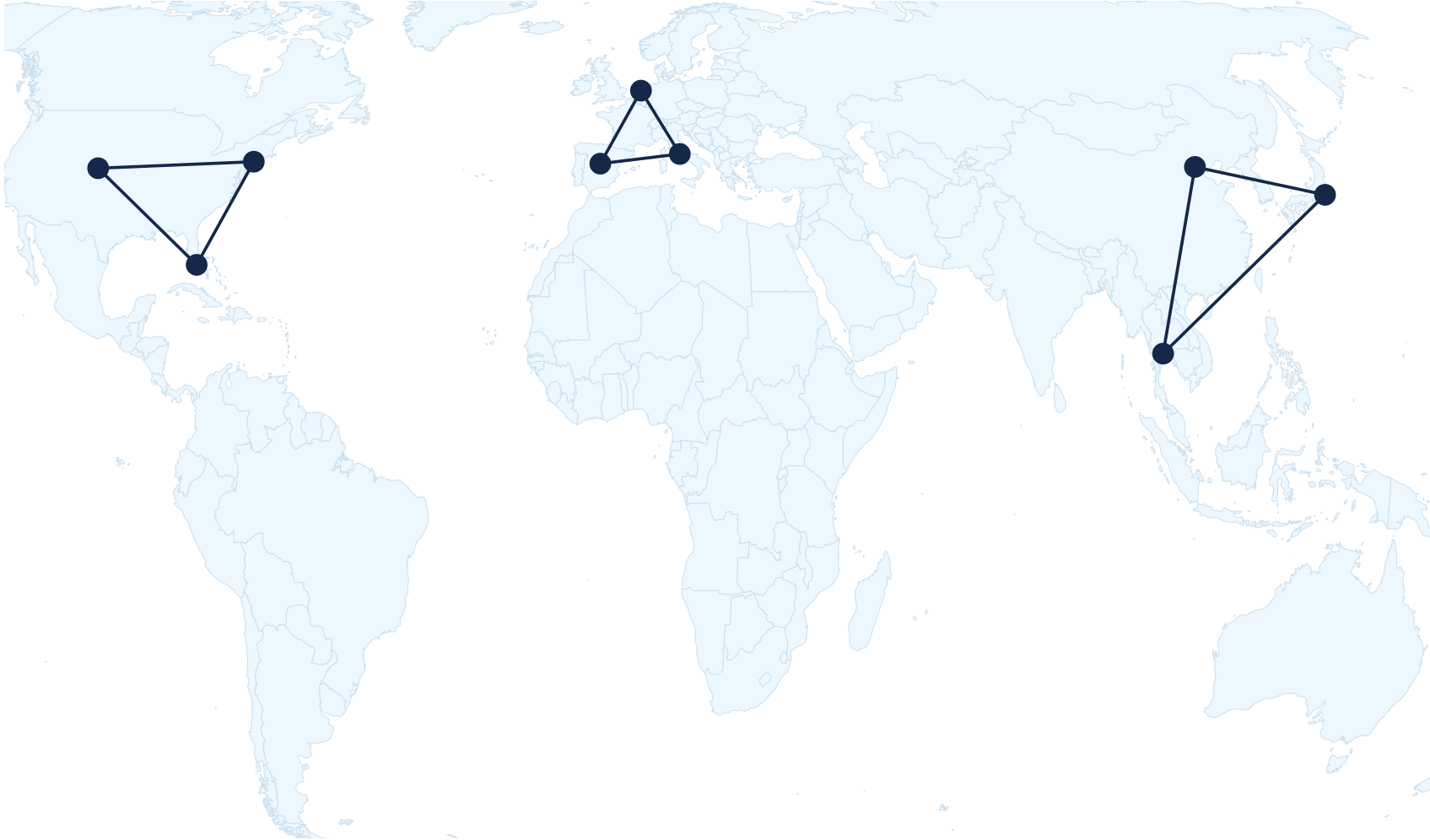
Replicate *within one region* for high availability



Replicate *across regions* for reads that follow the sun



Replicate *within separate regions* for localized data



CockroachDB

- Designed for global replication from the ground up
- Control where data is placed
- Bring the data to the user

Thank you

<https://www.cockroachlabs.com>

<https://cockroa.ch/cdnlessons>

Presented by Ben Darnell, CTO

