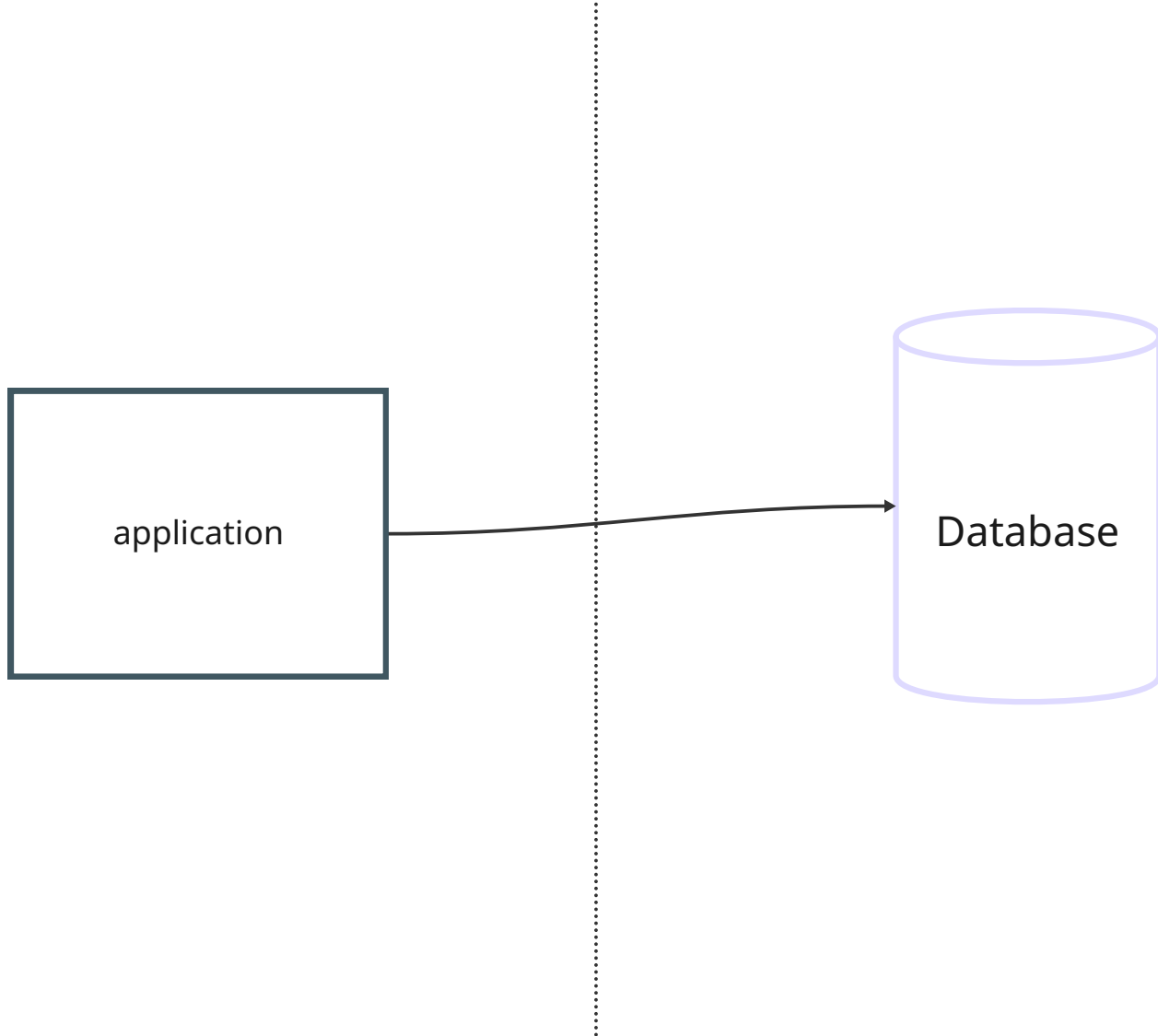


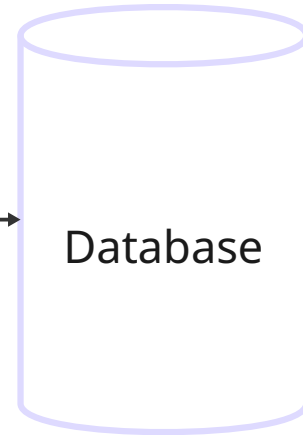
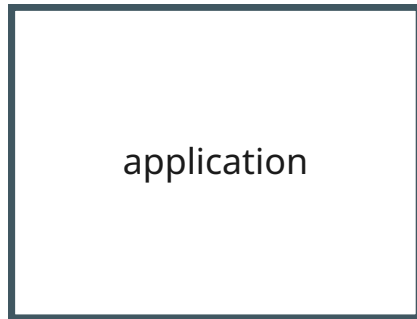
side-effects produce value

but persisting state is hard



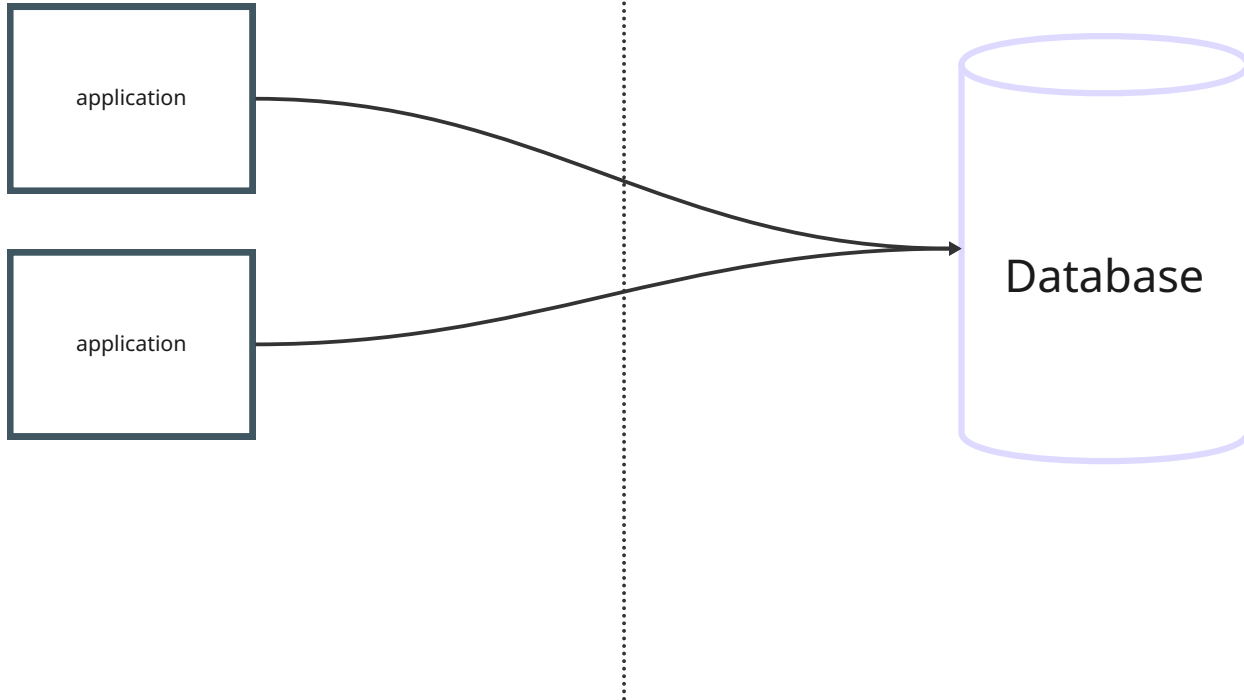
ephemeral

persisted

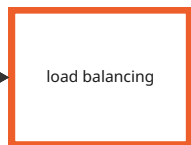
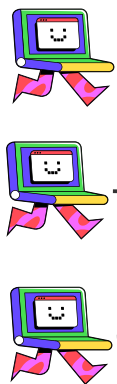


ephemeral

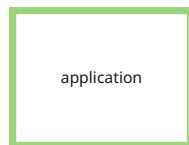
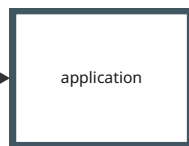
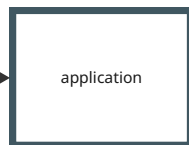
persisted



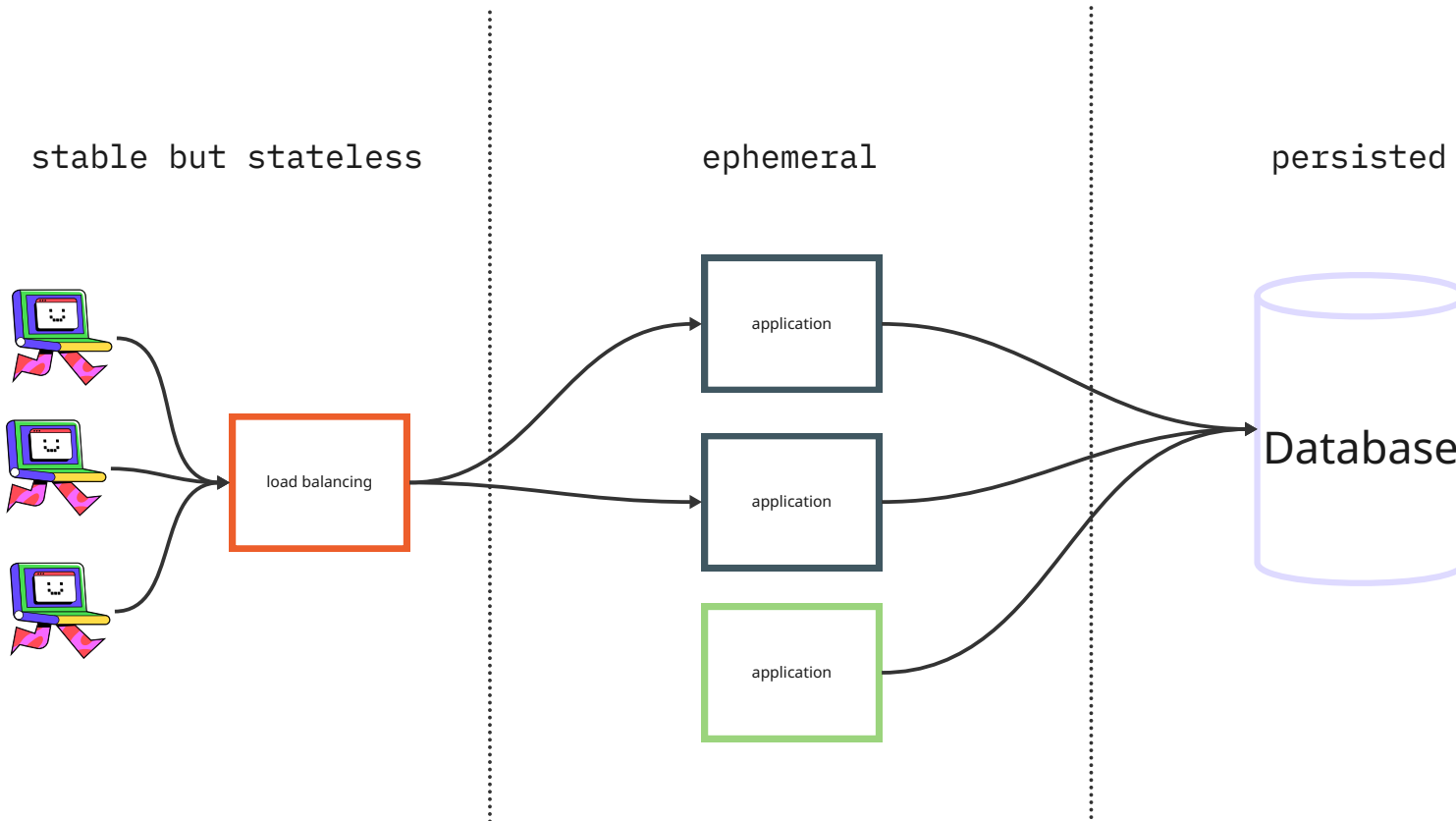
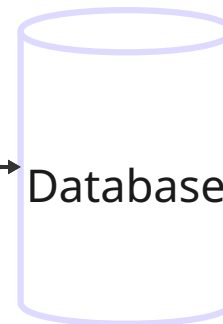
stable but stateless



ephemeral



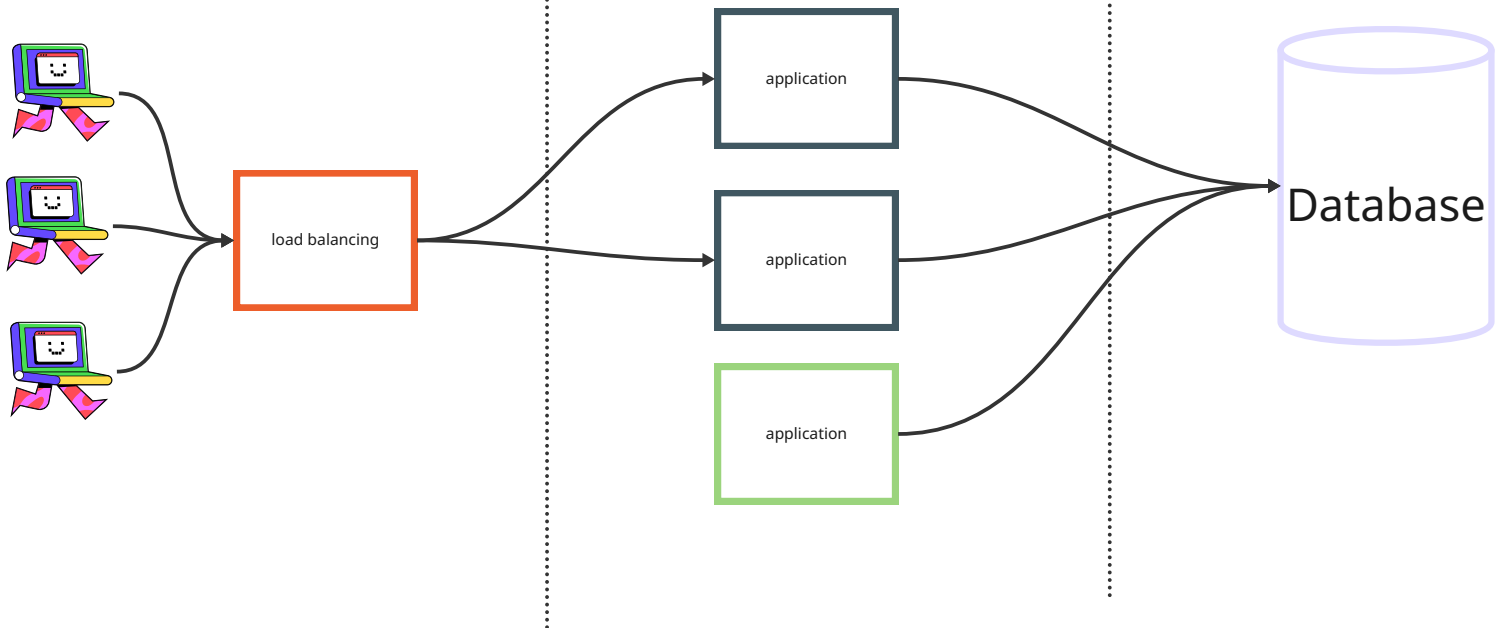
persisted



delegated to platform

the value we create

delegated to platform

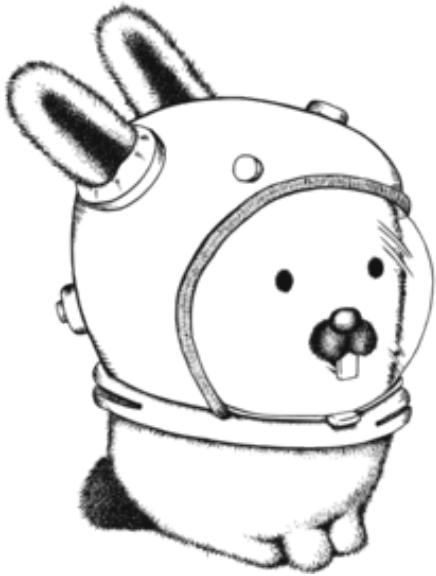


operations

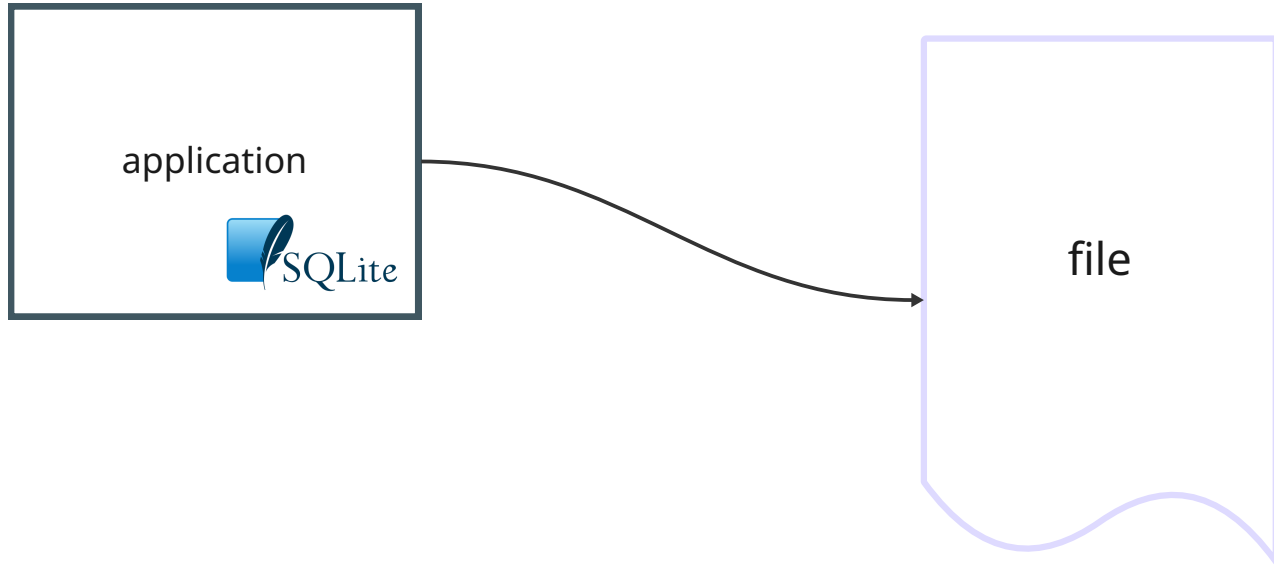


- creation
- authentication
- backups
- engine upgrades
- disaster recovery
- ...

what if everything is a file?



"machine"



- creation
- authentication
- backups
- engine upgrades
- disaster recovery
- ...

its code inside
your app

not needed

its a file

application
library

its a file

```

apiVersion: rds.services.k8s.aws/v1alpha1
kind: DBInstance
metadata:
  name: society-staging
spec:
  allocatedStorage: 20
  dbInstanceClass: db.t3.micro
  dbInstanceIdentifier: society-staging
  dbName: postgres
  port: 5432
  engine: postgres
  engineVersion: "13.18"
  masterUsername: "postgres"
  masterUserPassword:
    name: society--staging--secret-env-vars
    key: PGPASSWORD
  vpcSecurityGroupIDs:
    - sg-0a04b1c8433227e63
  dbSubnetGroupName:
    elife-flux-prod-ack-rds-controller-all-dbs
---
apiVersion: services.k8s.aws/v1alpha1
kind: FieldExport
metadata:
  name: society-staging-dbinstance-endpoint-address
  namespace: society
spec:
  to:
    name: society--staging--public-env-vars
    kind: configmap
    key: PGHOST
  from:
    path: ".status.endpoint.address"
    resource:
      group: rds.services.k8s.aws
      kind: DBInstance
      name: society-staging

```

```
const db = new sqlite3.Database('application-state.db');
```

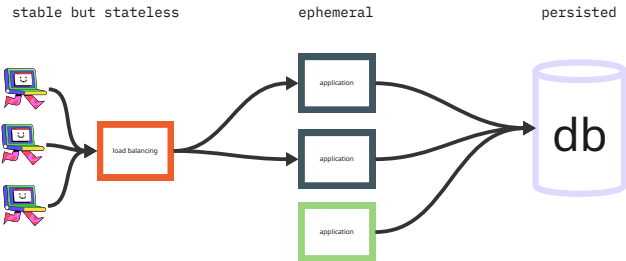
- exploring prod
- local dev
- tests
- speed of light
- ...

its a file

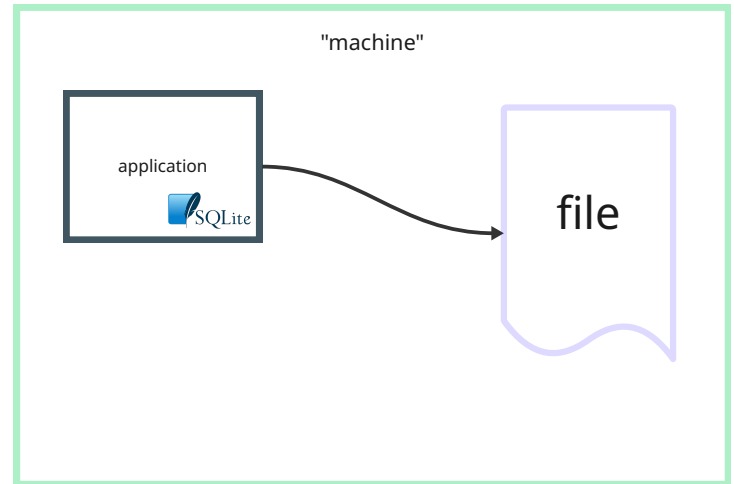
its a file

```
const db = new sqlite3.Database(':memory:');
```

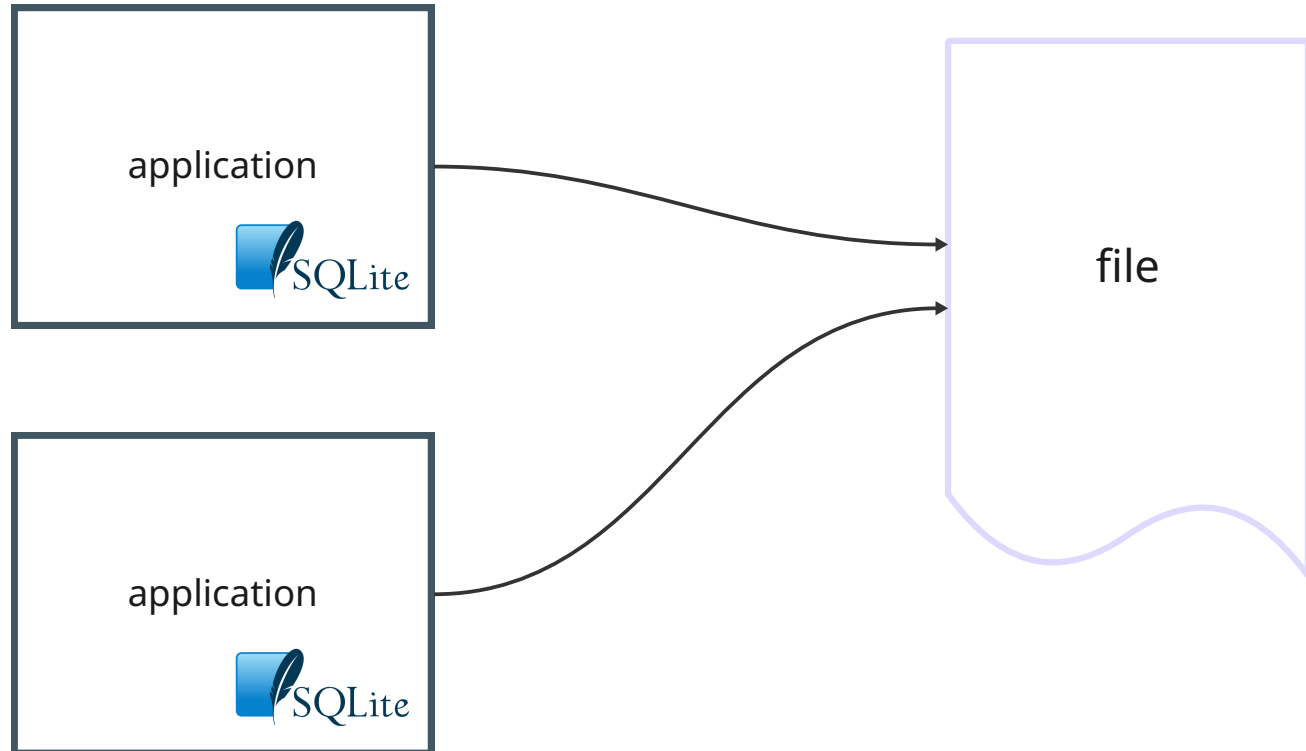
in the same
process



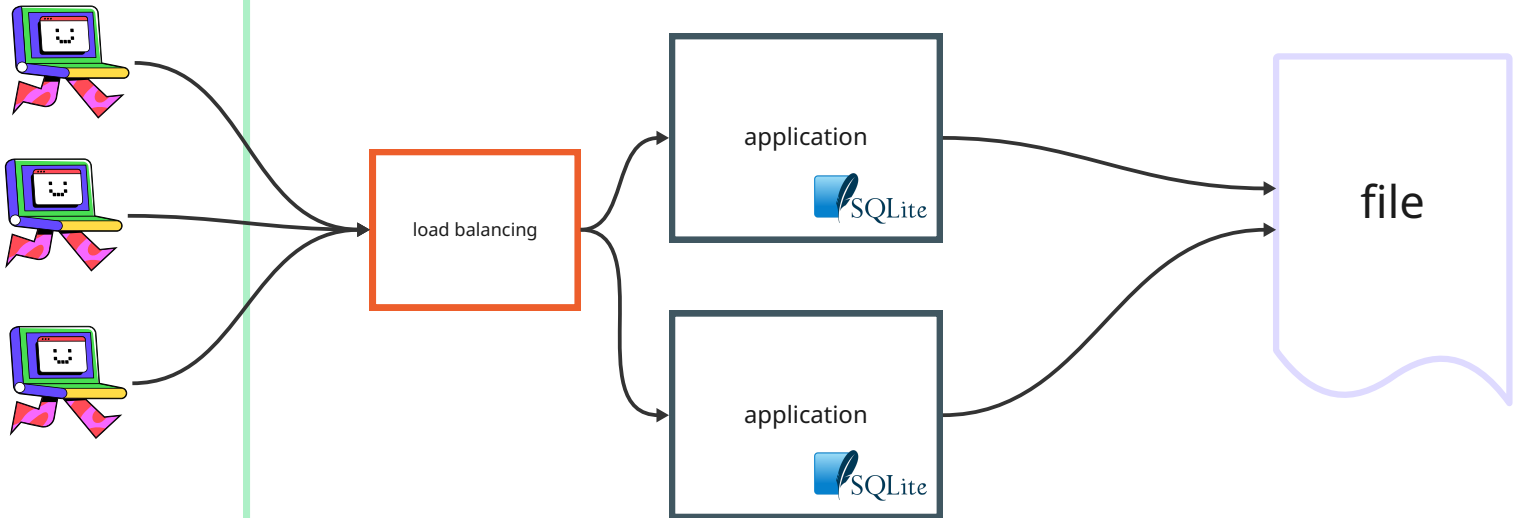
- rolling release?
- scaling?
- durability?



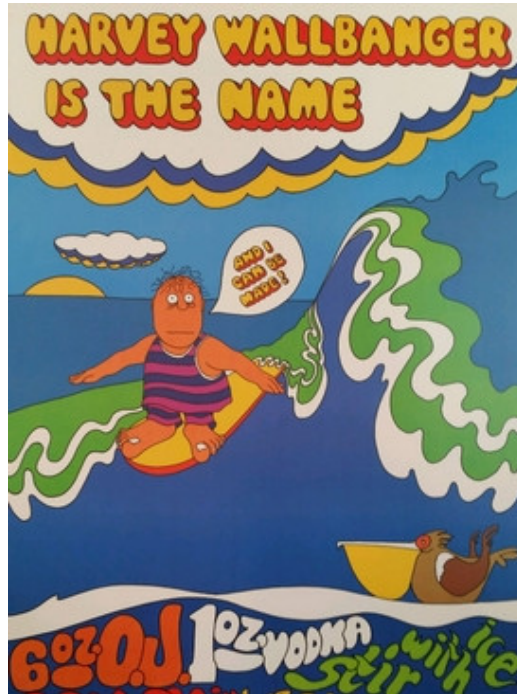
operating system



operating system



demo time



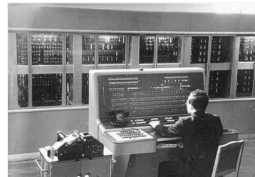
rolling releases?

yes, within single machine/OS

rolling releases?

yes, within single machine/OS

scaling?

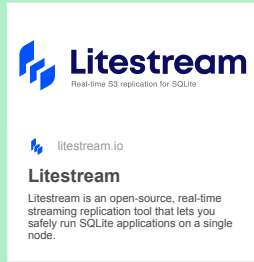


 use.expensify.com

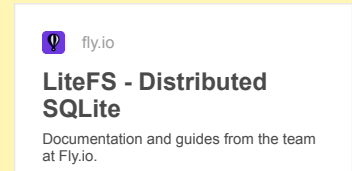
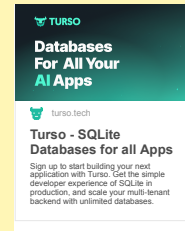
Scaling SQLite to 4M QPS on a Single Server (EC2 vs Bare Metal)

Discover how Expensify achieved remarkable scaling: 4M QPS on one server! Dive into optimizing SQLite for impressive performance gains.

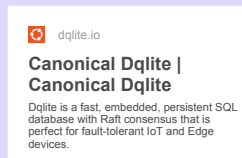
only durability

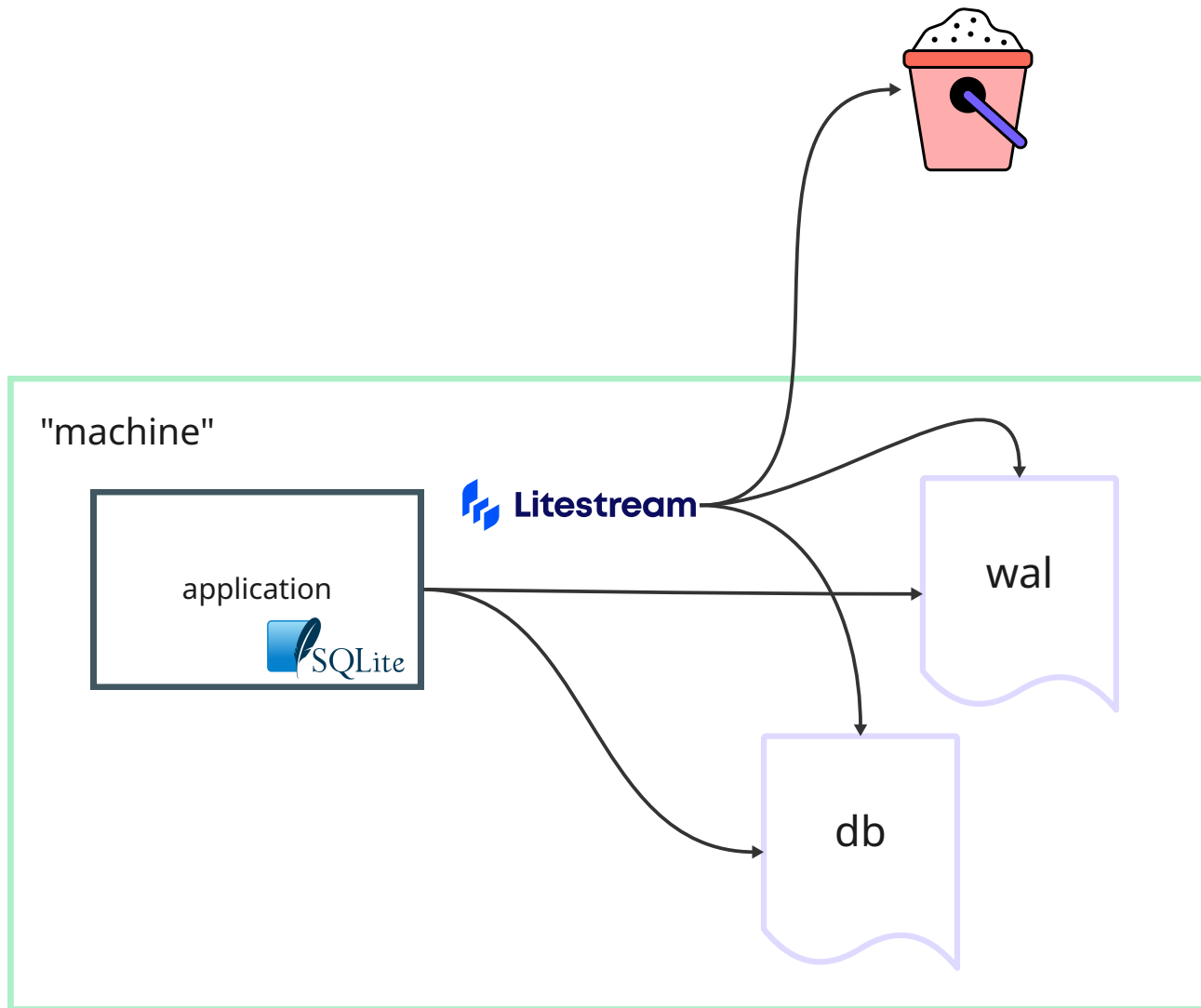


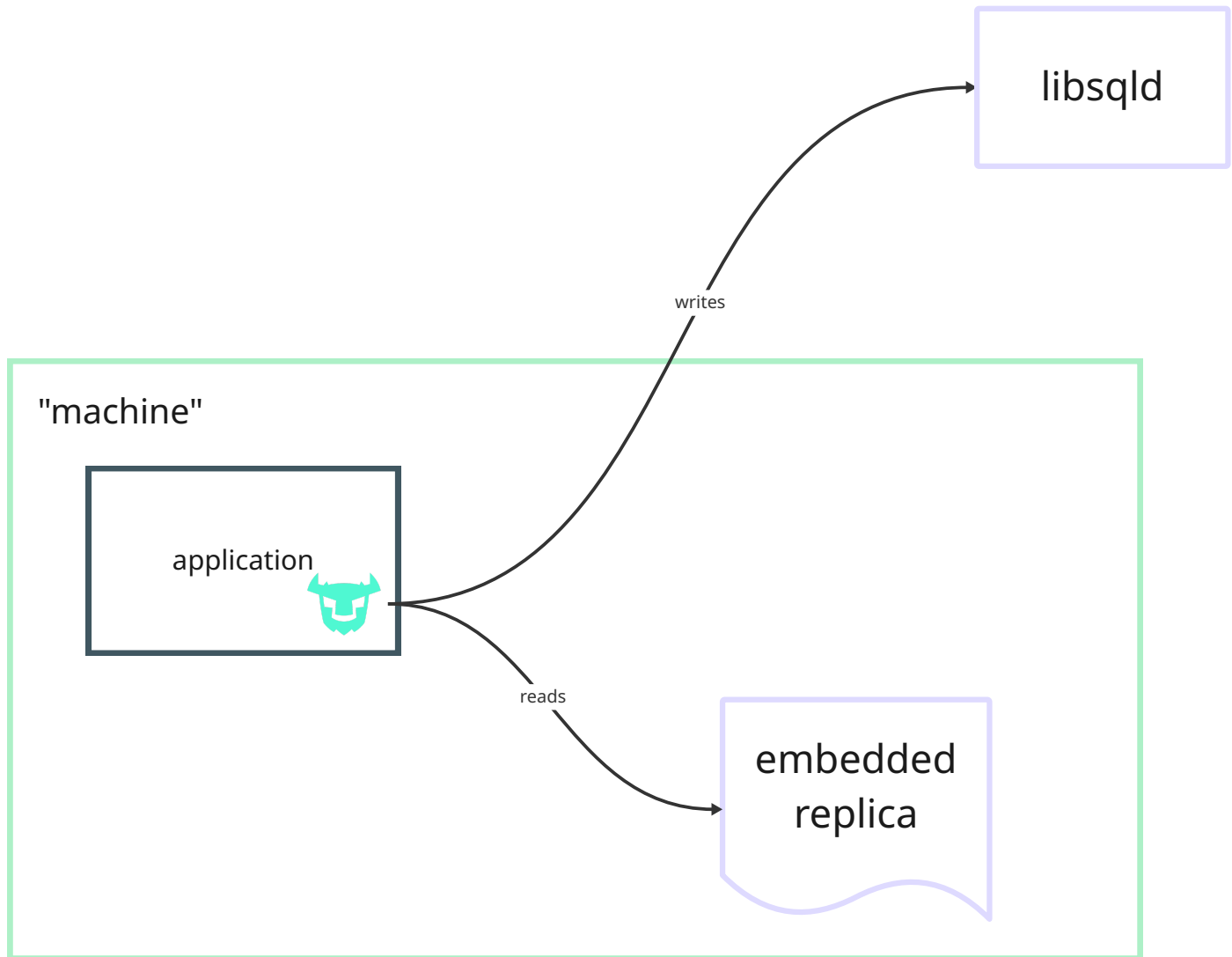
single writer with
read replicas

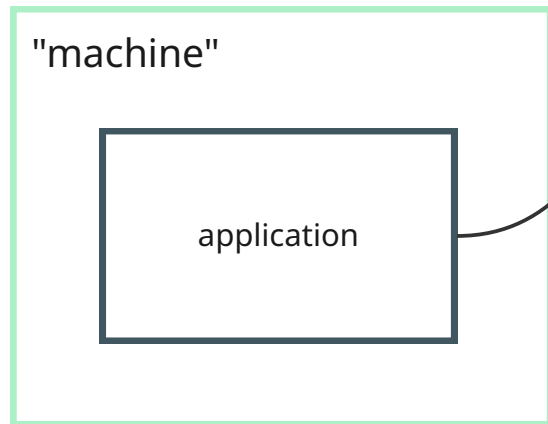


distributed sqlite









rqlite



rqlite



rqlite



