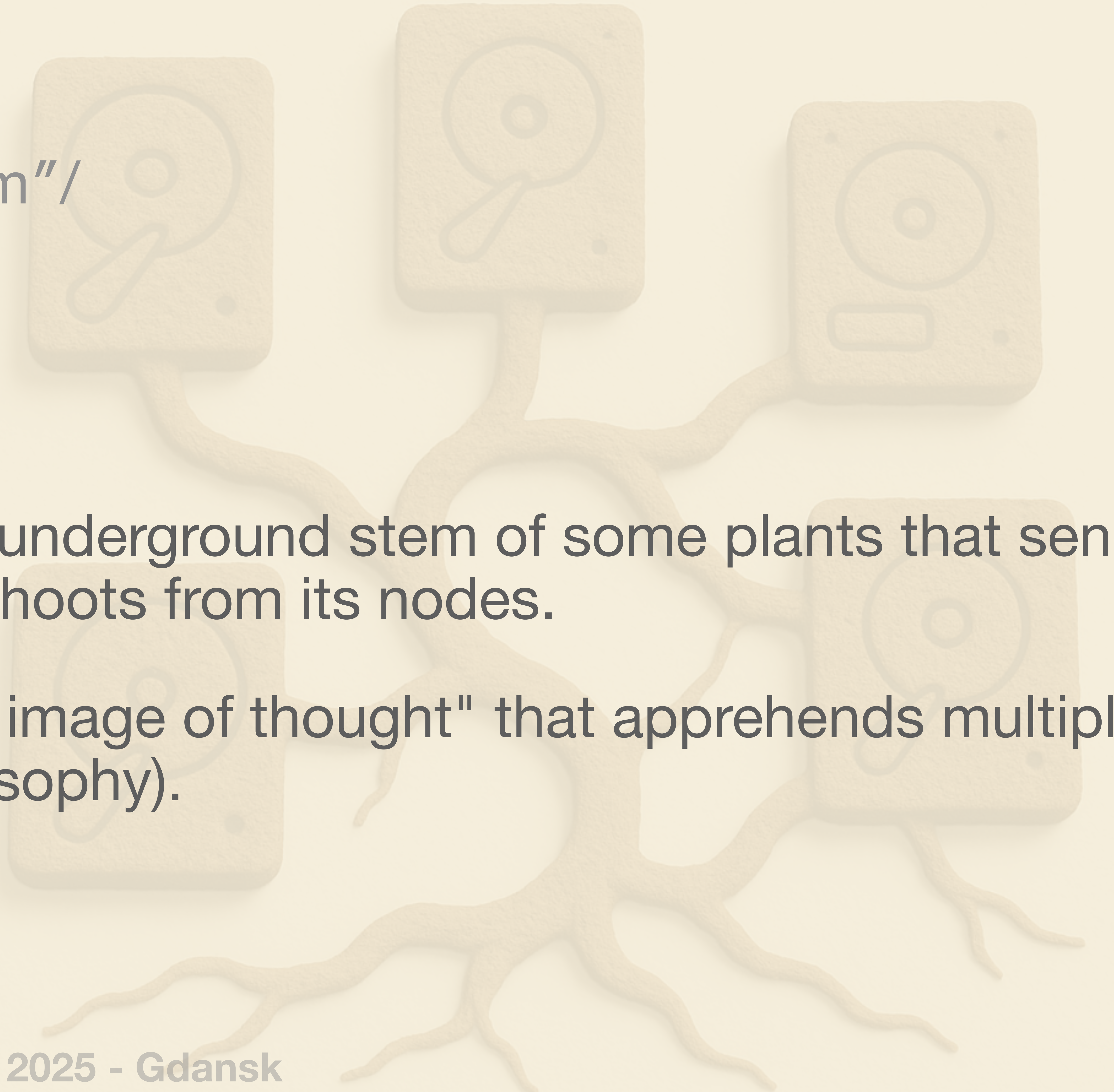


rhizome /rī′zōm″/

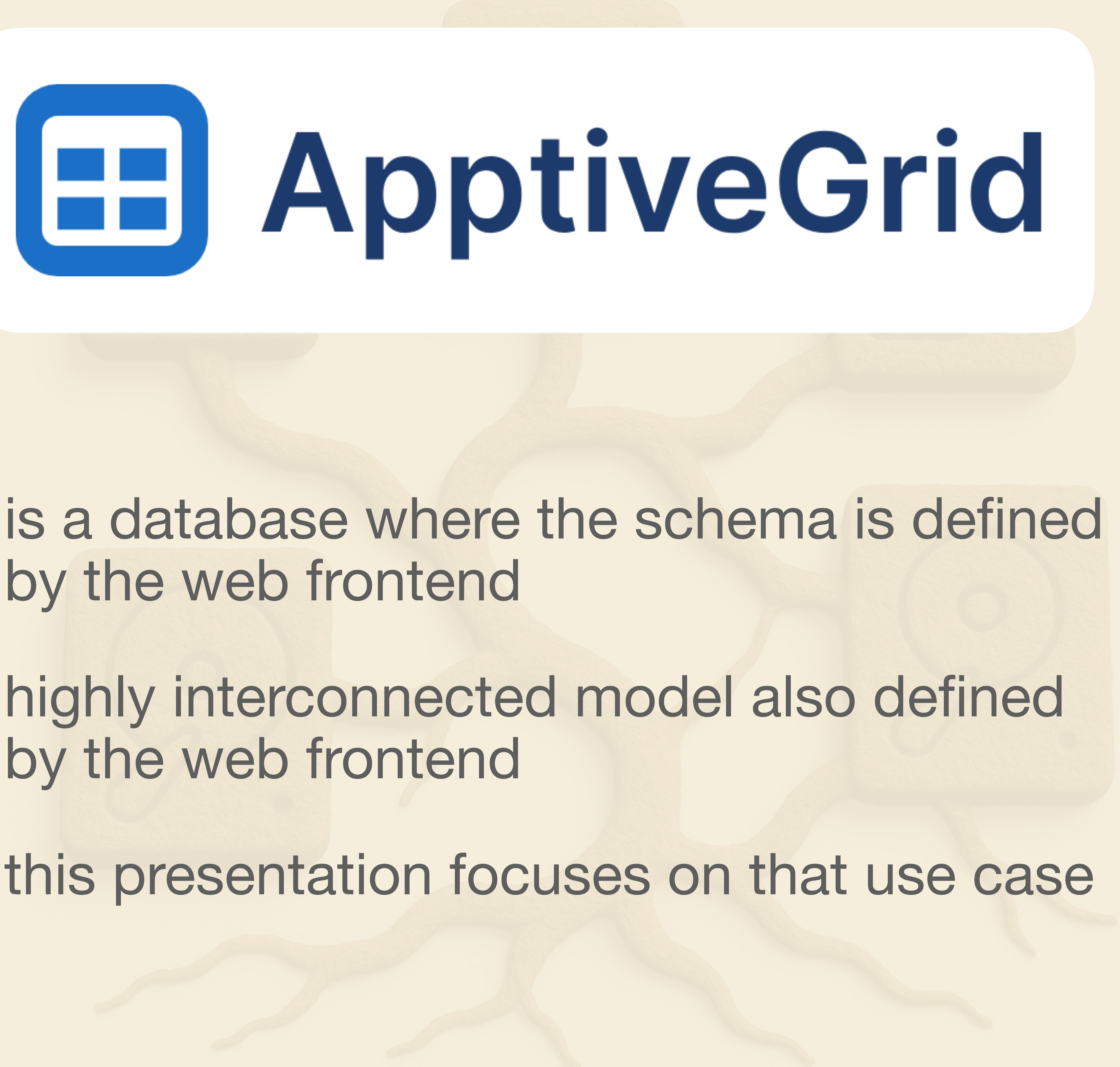
noun

1. A horizontal underground stem of some plants that sends out roots and shoots from its nodes.

2. A so-called "image of thought" that apprehends multiplicities. See Rhizome (philosophy).

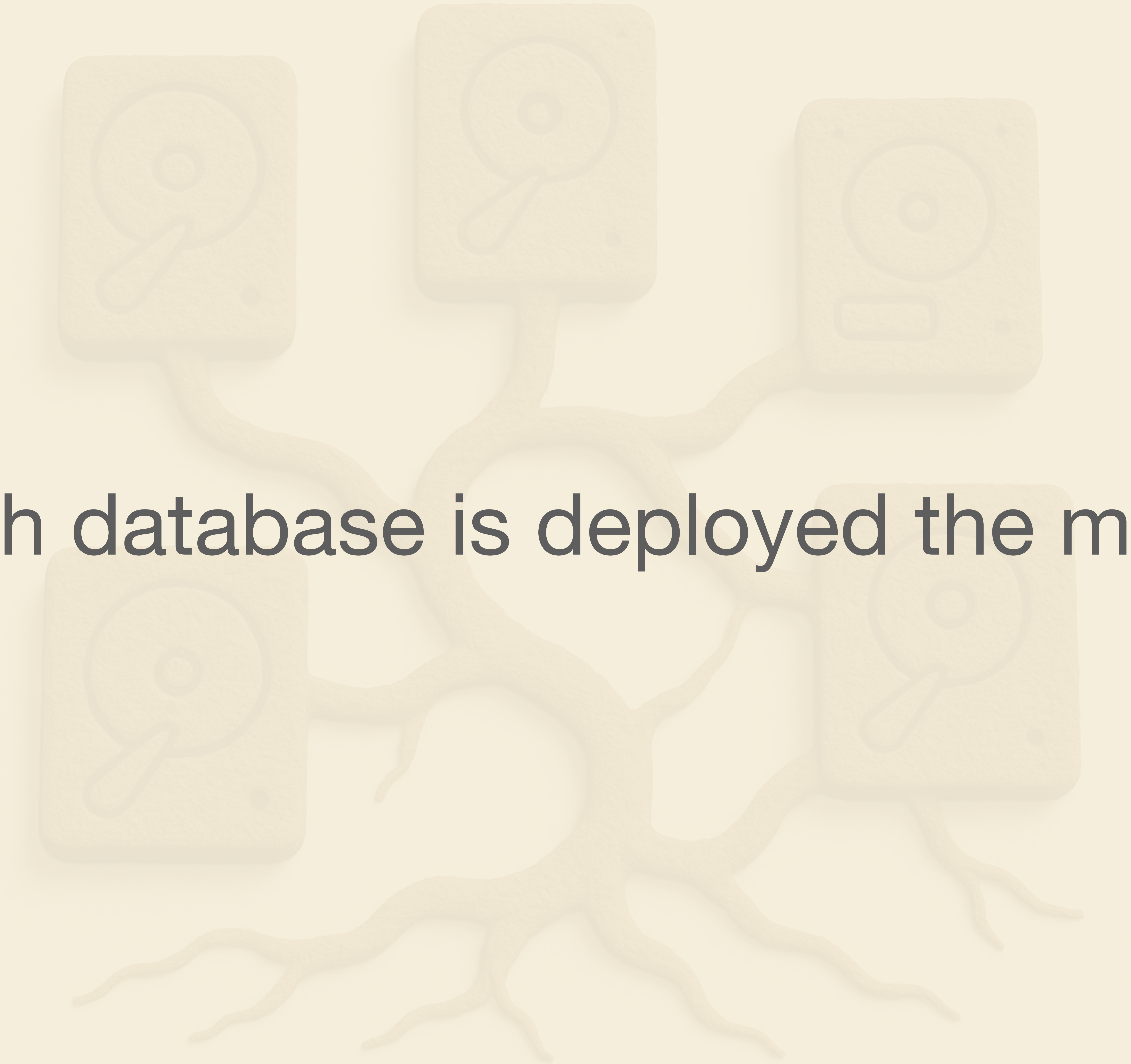
...

Currently I work for the LAPD
(prompting not included)



ApptiveGrid

- is a database where the schema is defined by the web frontend
- highly interconnected model also defined by the web frontend
- this presentation focuses on that use case



Which database is deployed the most?

Which database is deployed the most?

SQLite

The good part

SQLite

The good part

It's a library!

SQLite

The bad part

SQLite

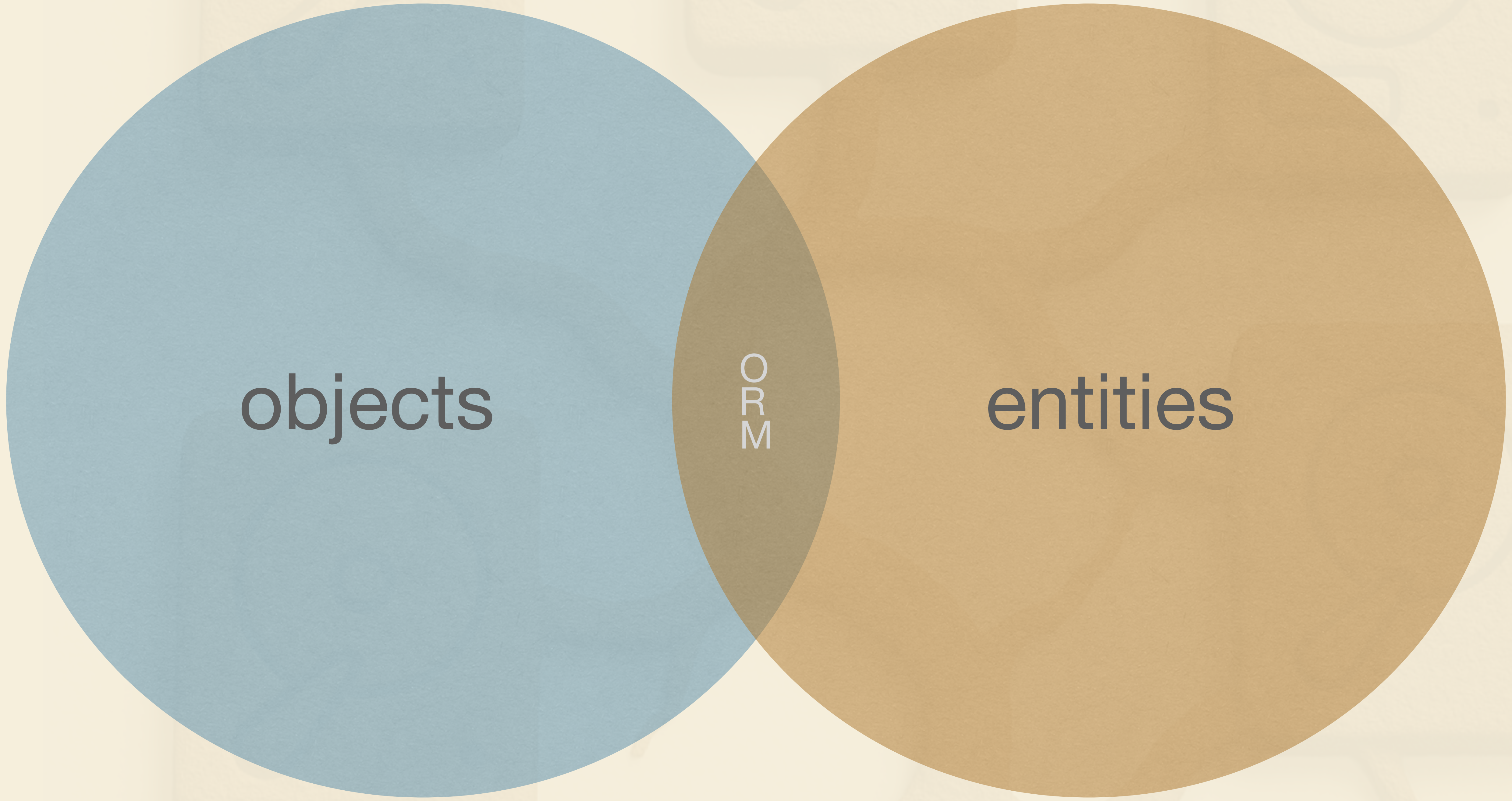
The bad part

Is SQL bad?

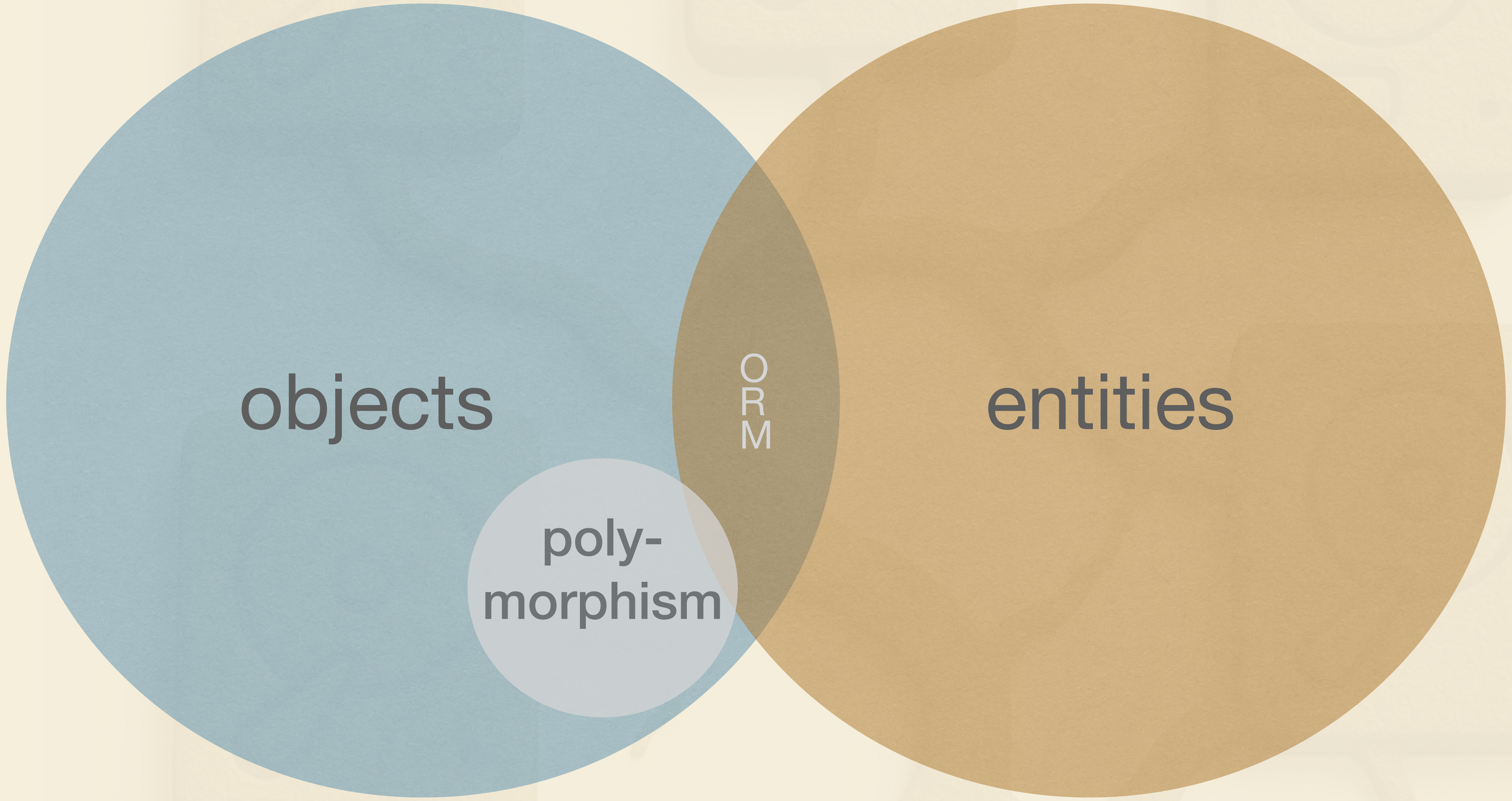
Is SQL bad?

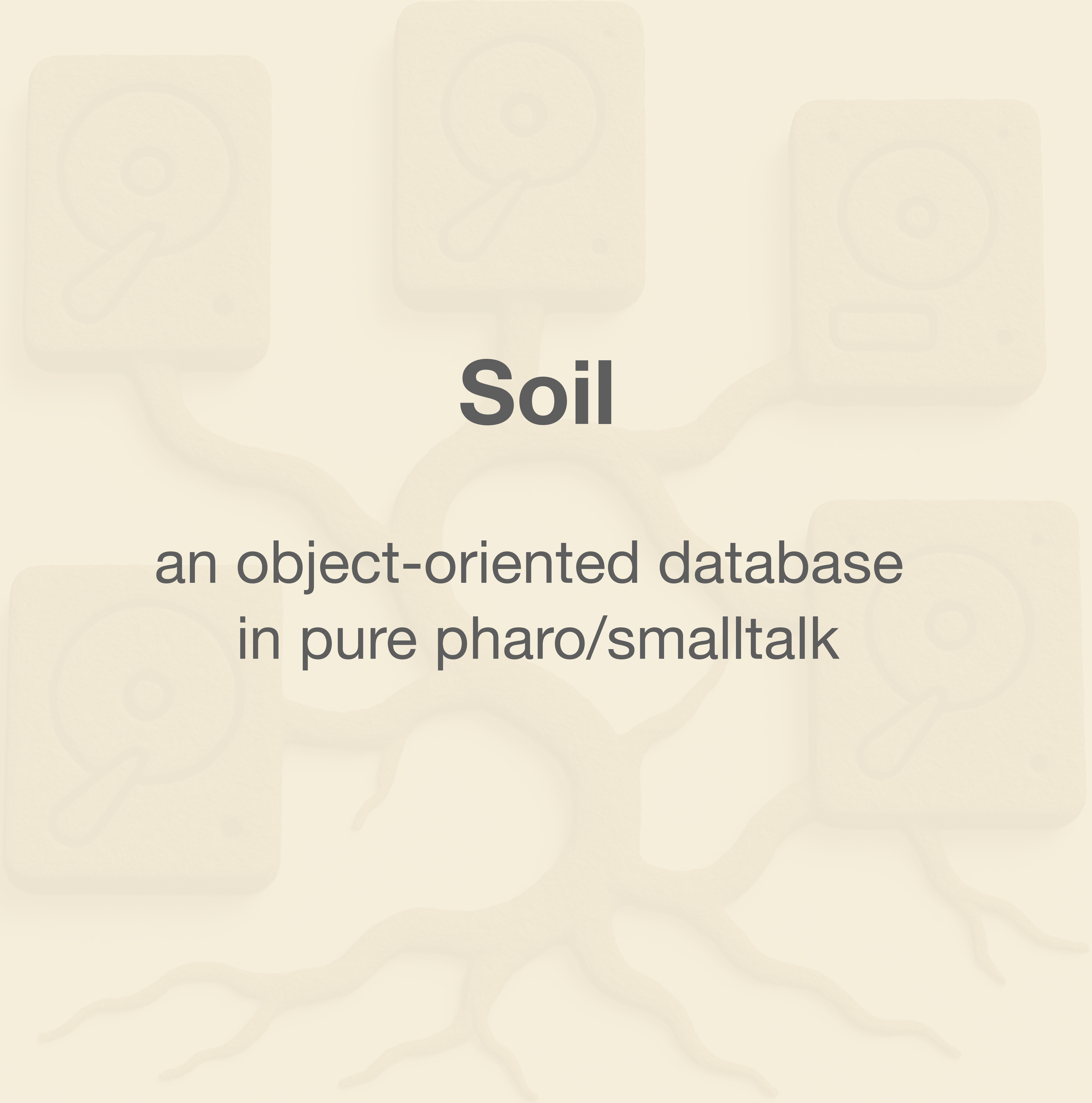
No!!..but...

Entities are not objects!



Entities are not objects!

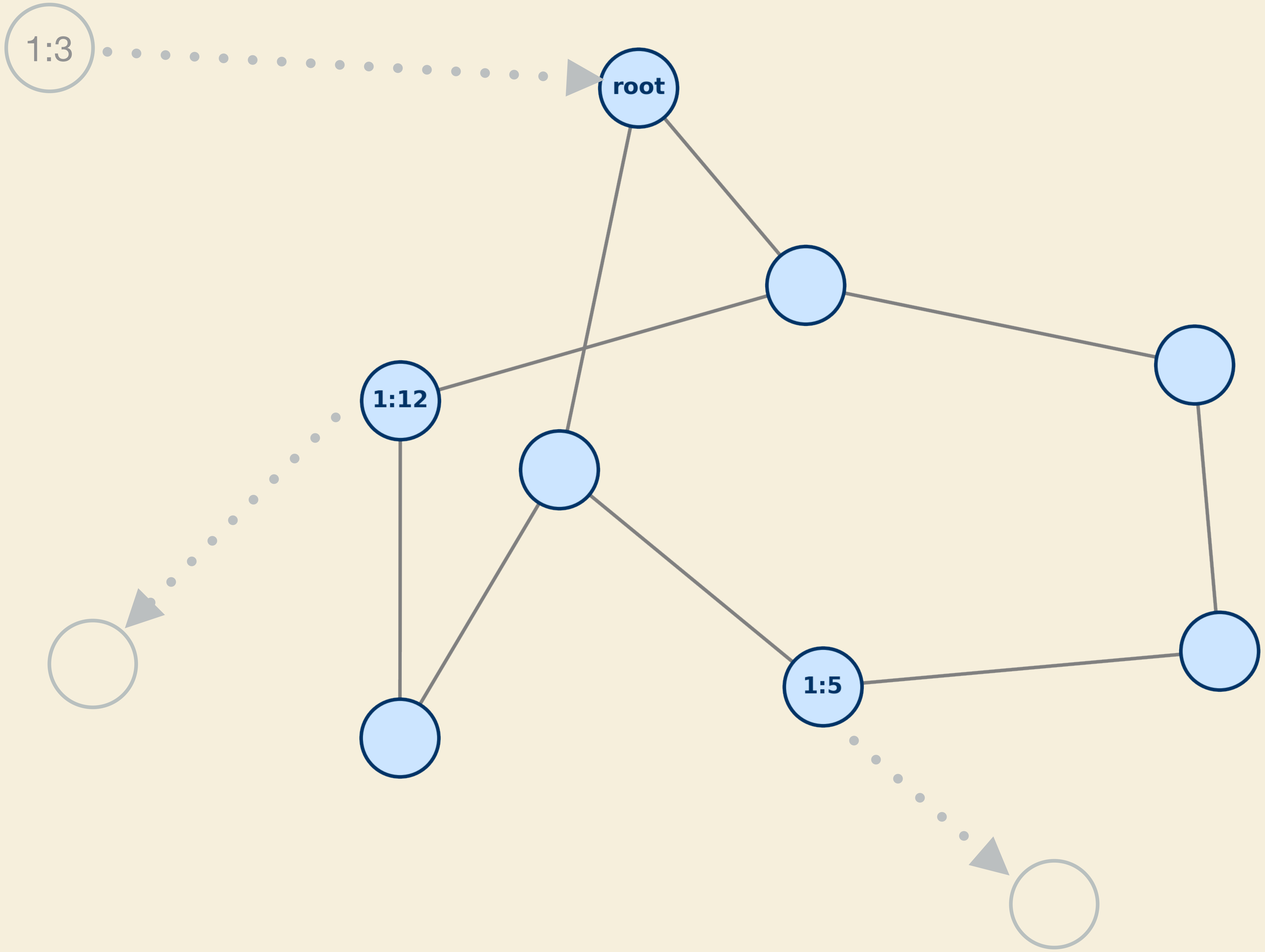




Soil

an object-oriented database
in pure pharo/smalltalk

Soil - Clusters

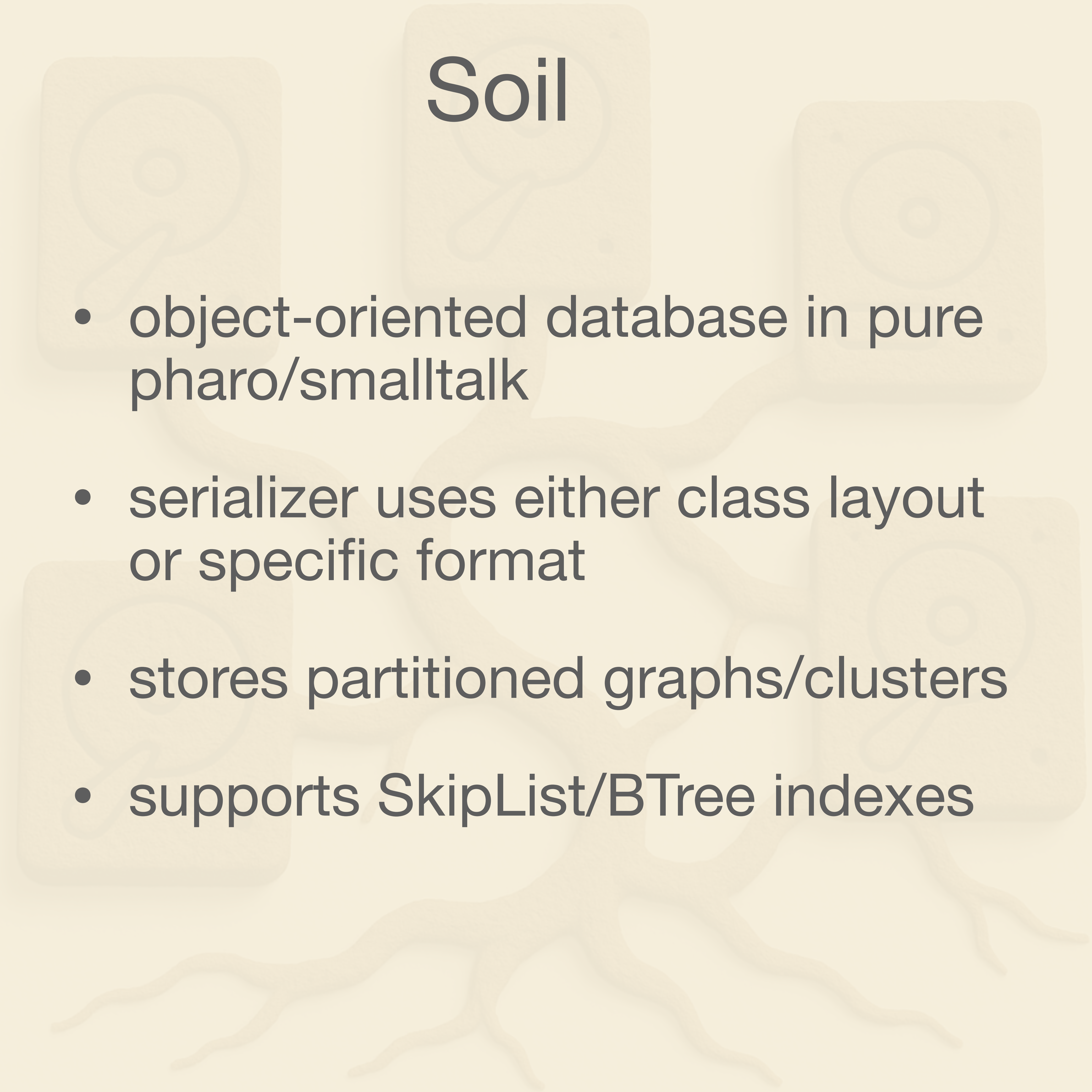


Soil - Serializer

```
Object>>#soilBasicSerialize: serializer

    self class classLayout
        soilBasicSerialize: self
        with: serializer
```

```
String>>#soilBasicSerialize: serializer

    serializer nextPutString: self
```

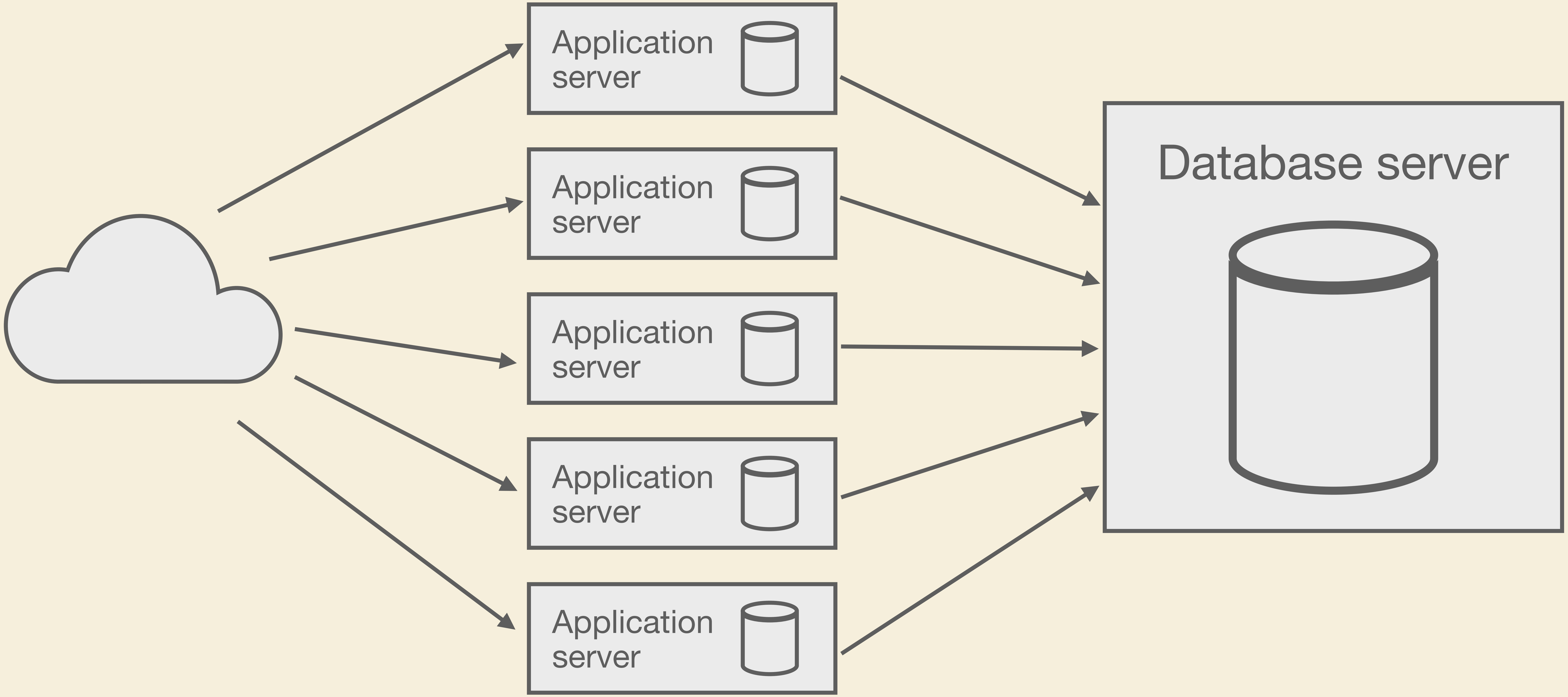
Soil



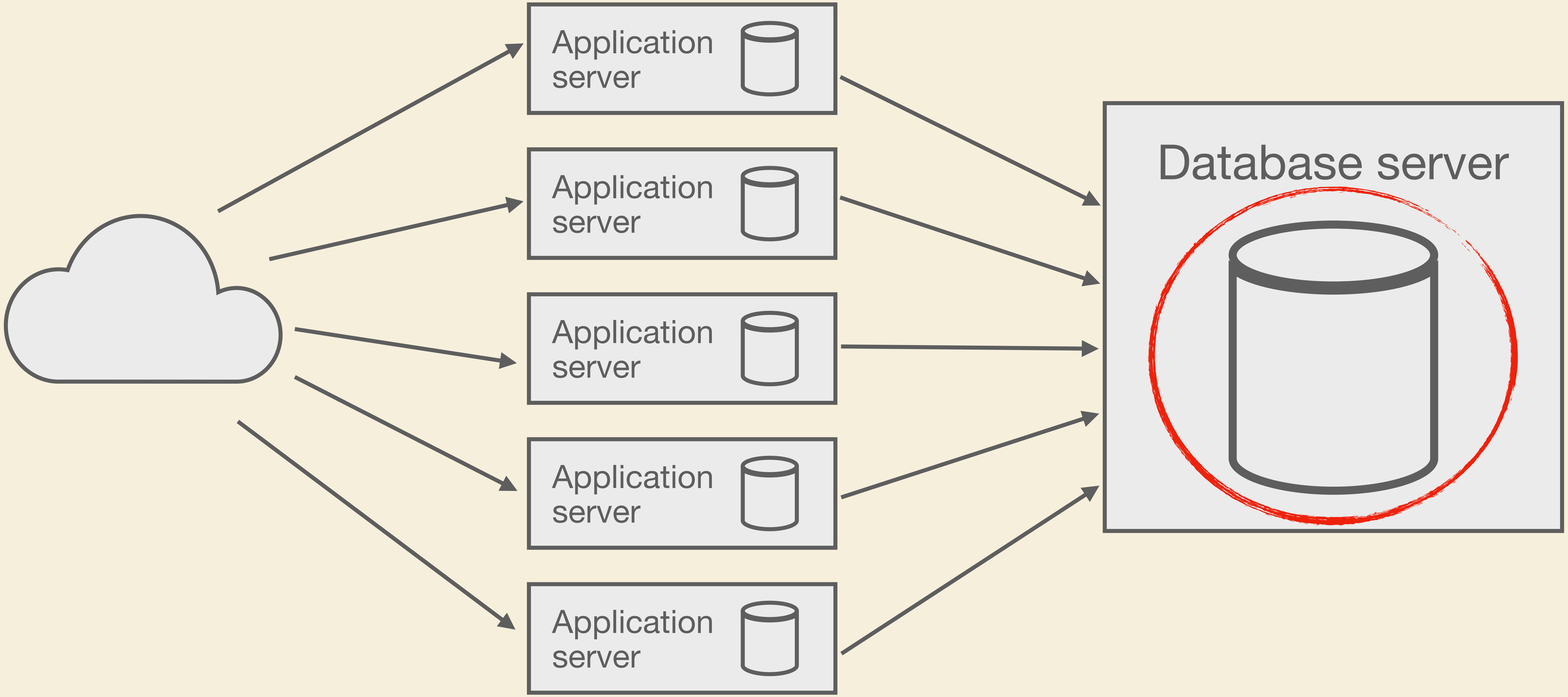
- object-oriented database in pure pharo/smalltalk
- serializer uses either class layout or specific format
- stores partitioned graphs/clusters
- supports SkipList/BTree indexes

"*But it can only store data on single machine!*"

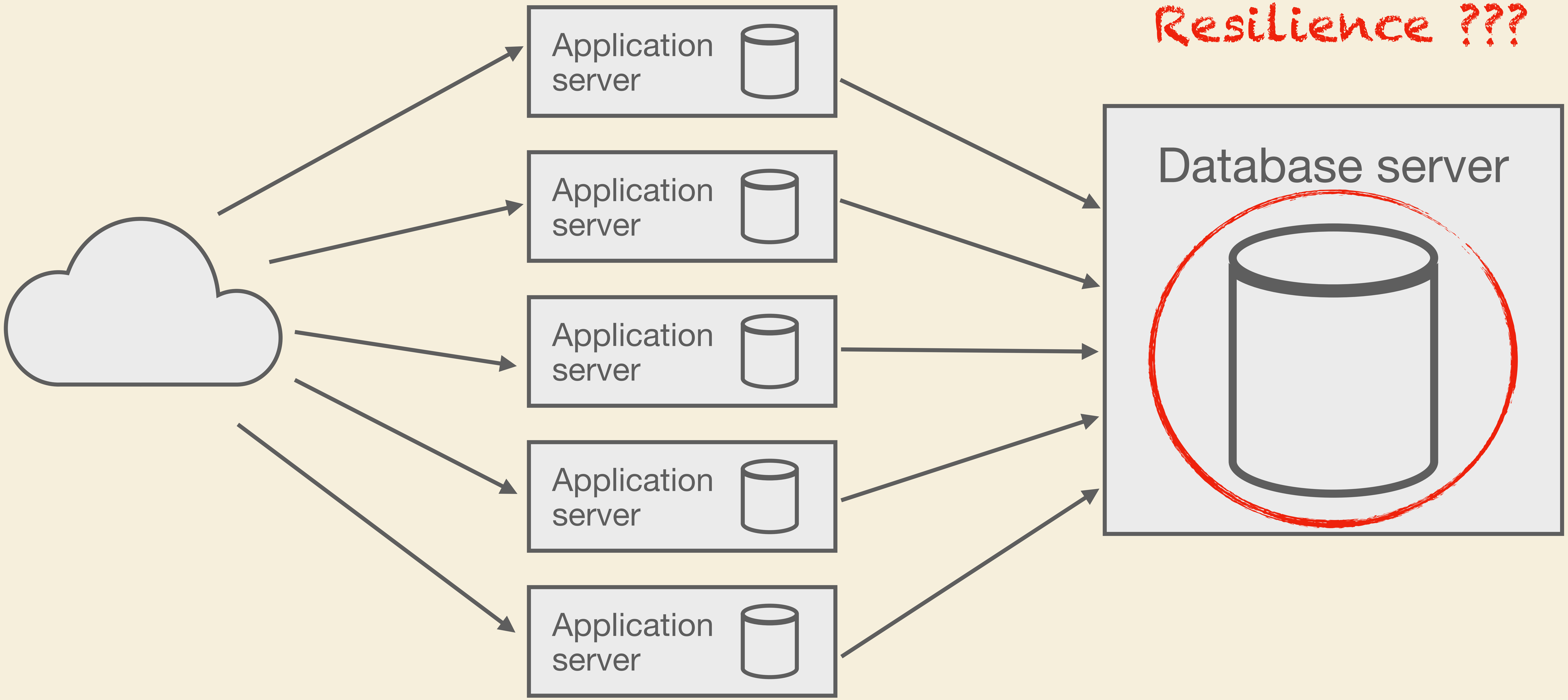
Being 3-tiered



Being 3-tiered



Being 3-tiered



Resilience ???

Application server

Application server

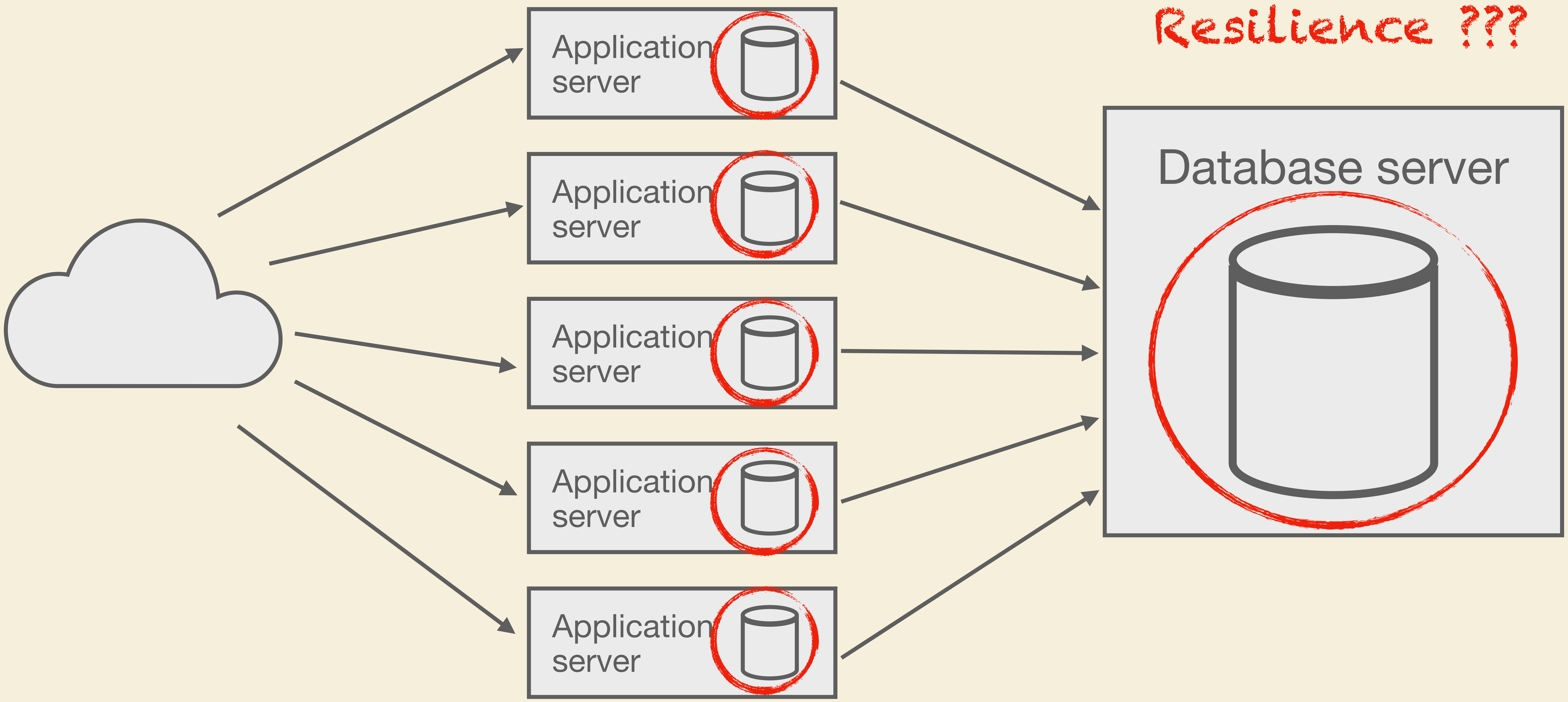
Application server

Application server

Application server

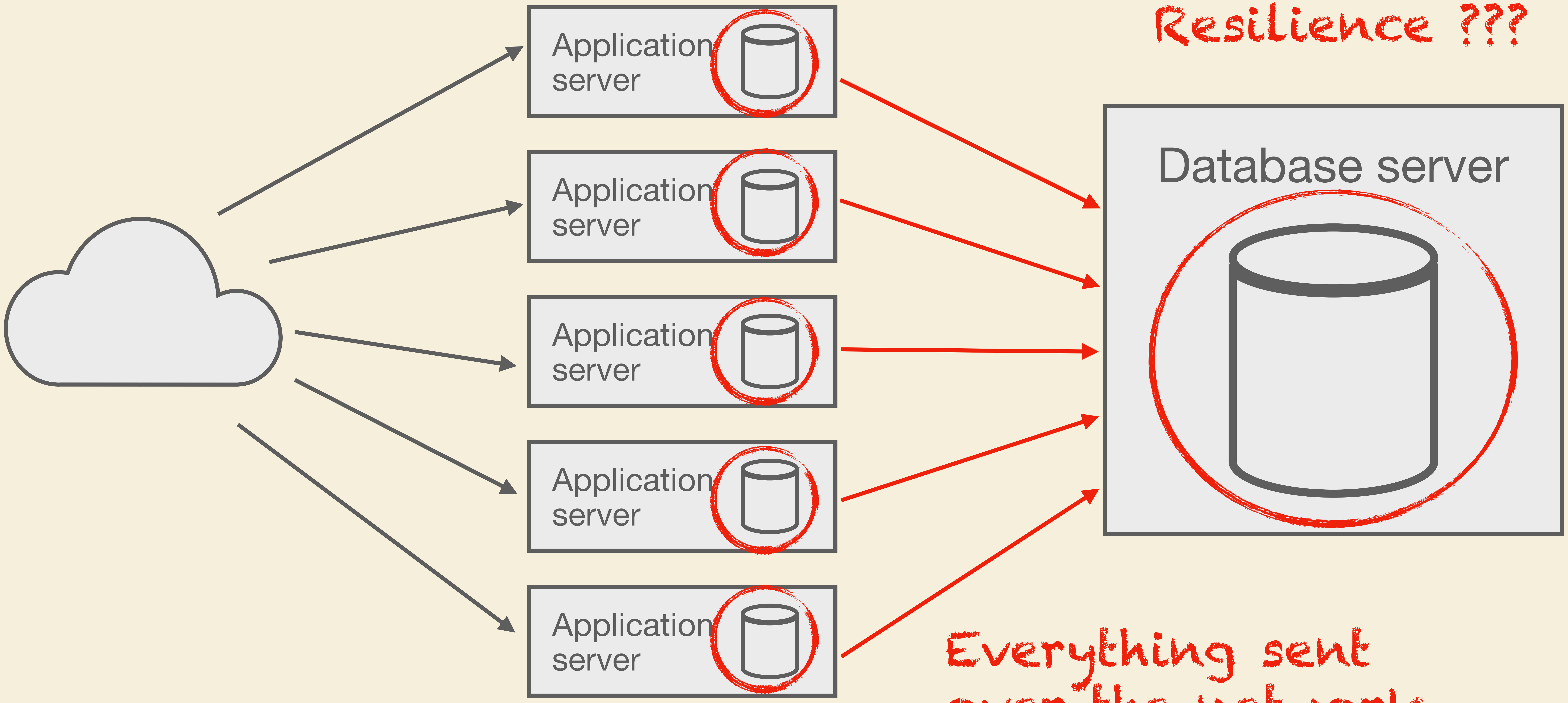
Database server

Being 3-tiered



Resilience ???

Being 3-tiered

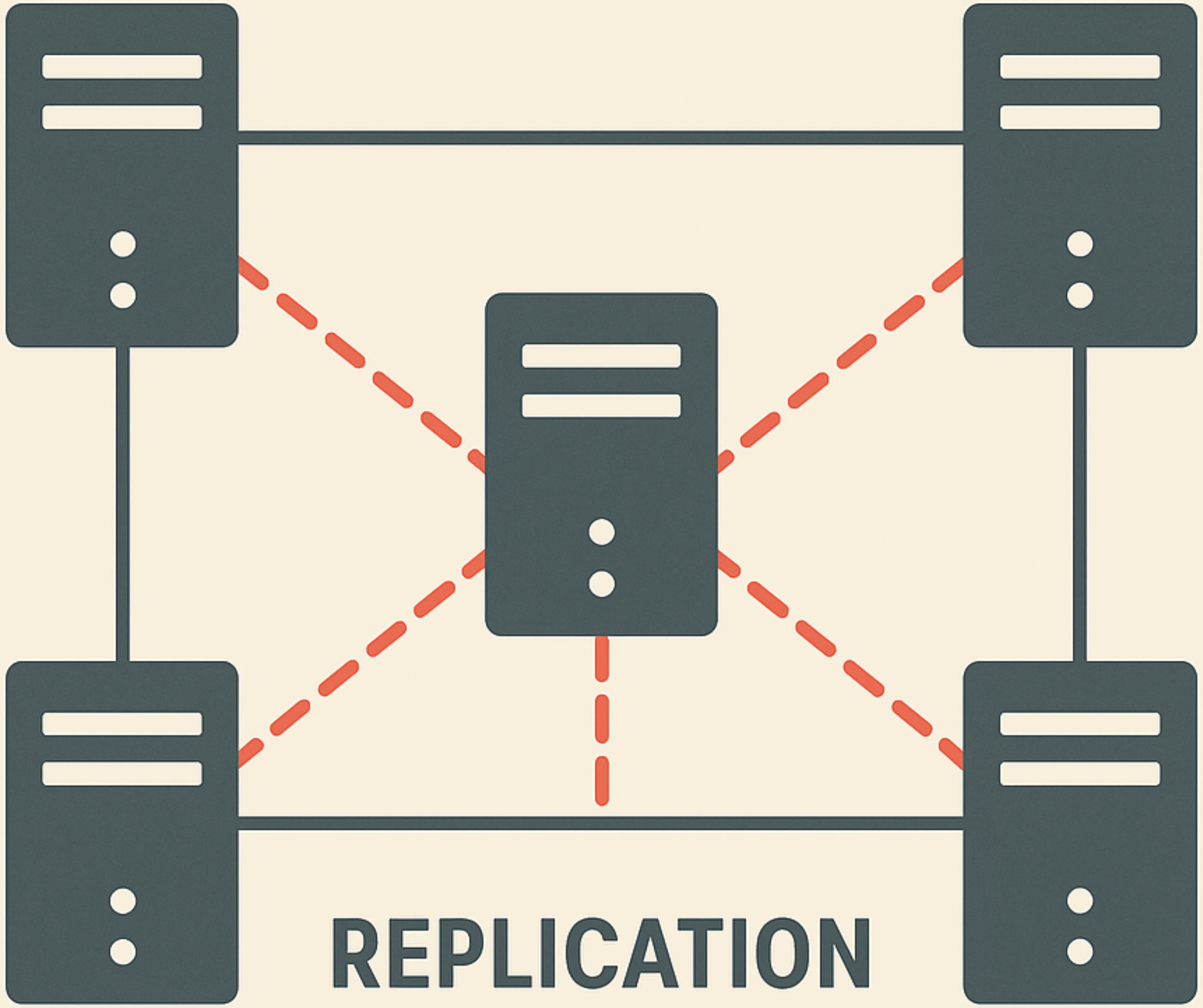


Resilience ???

Everything sent over the network

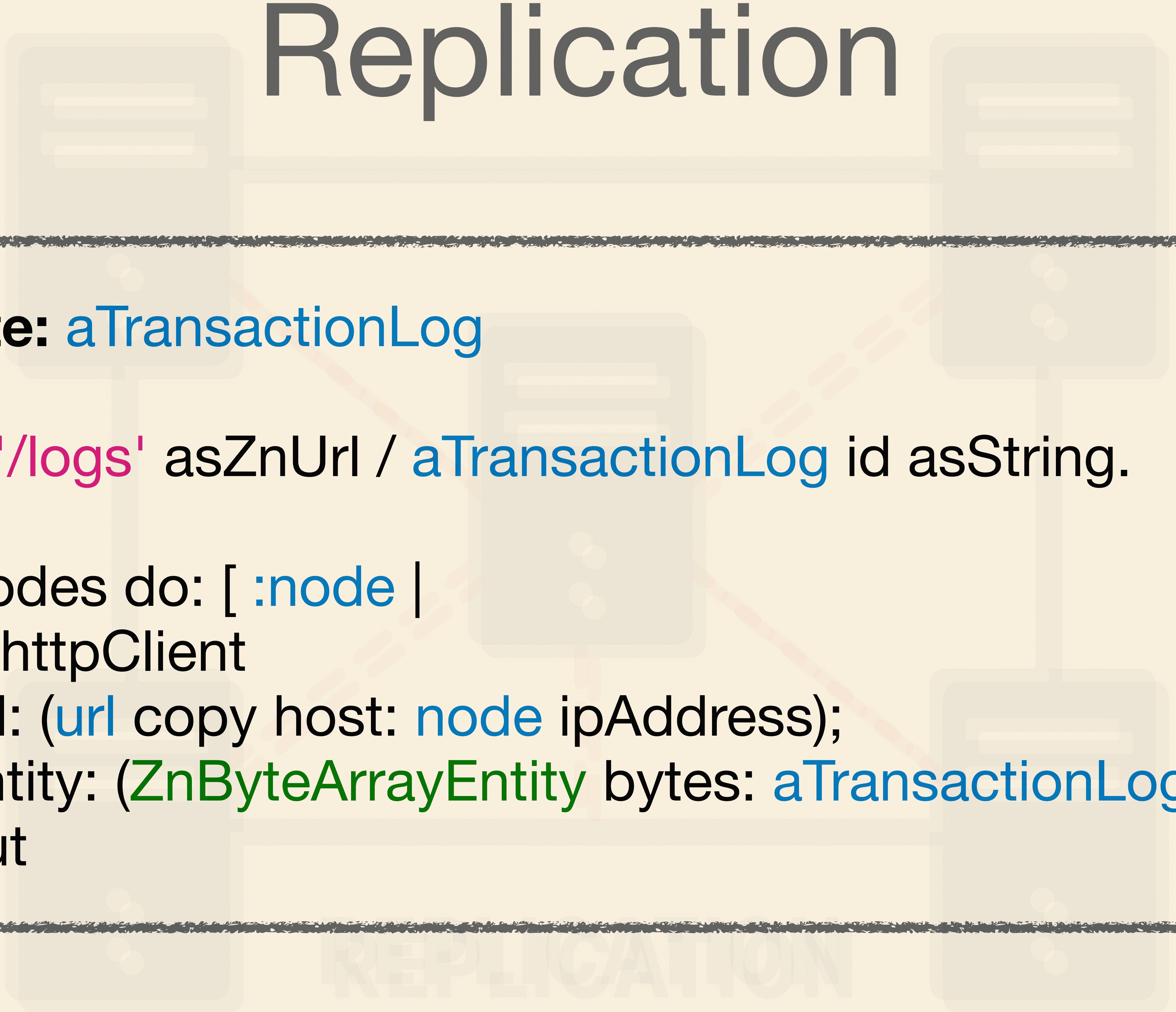
the goal: distributed soil

- avoid central components
- utilize all disks for I/O performance
- enable horizontal scaling for resilience and performance



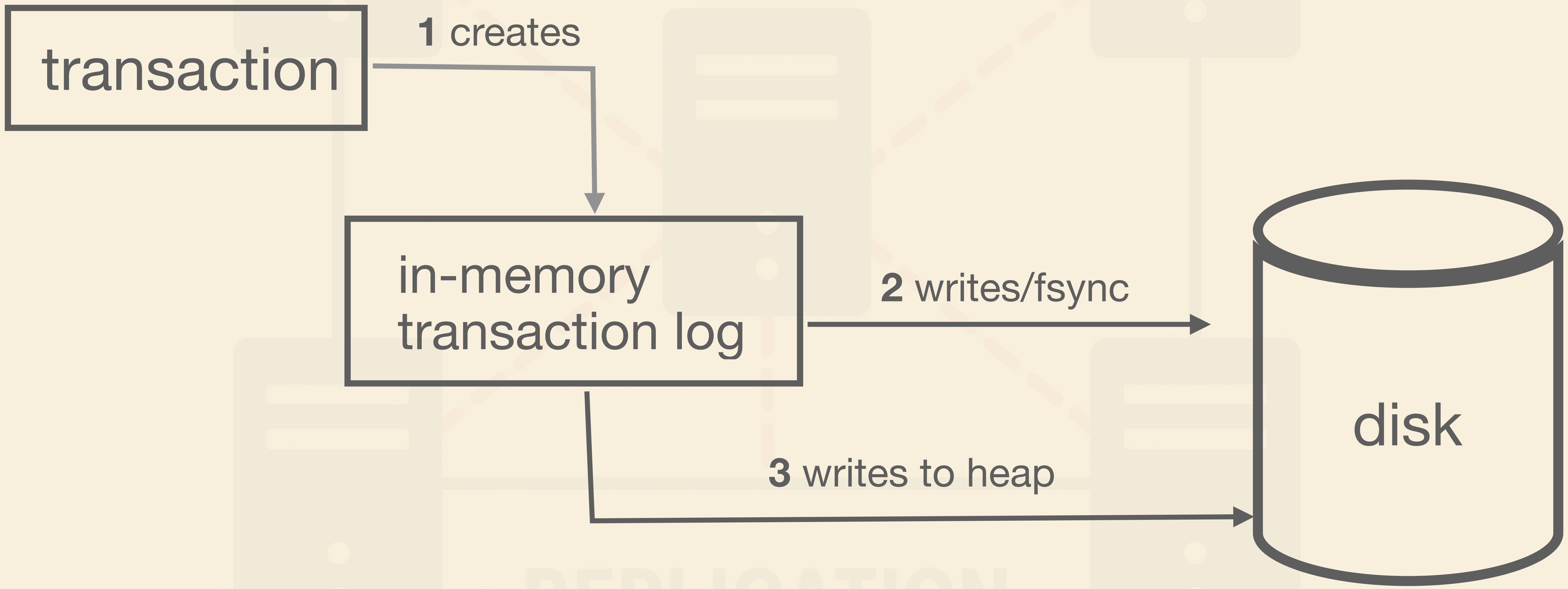
REPLICATION

Replication

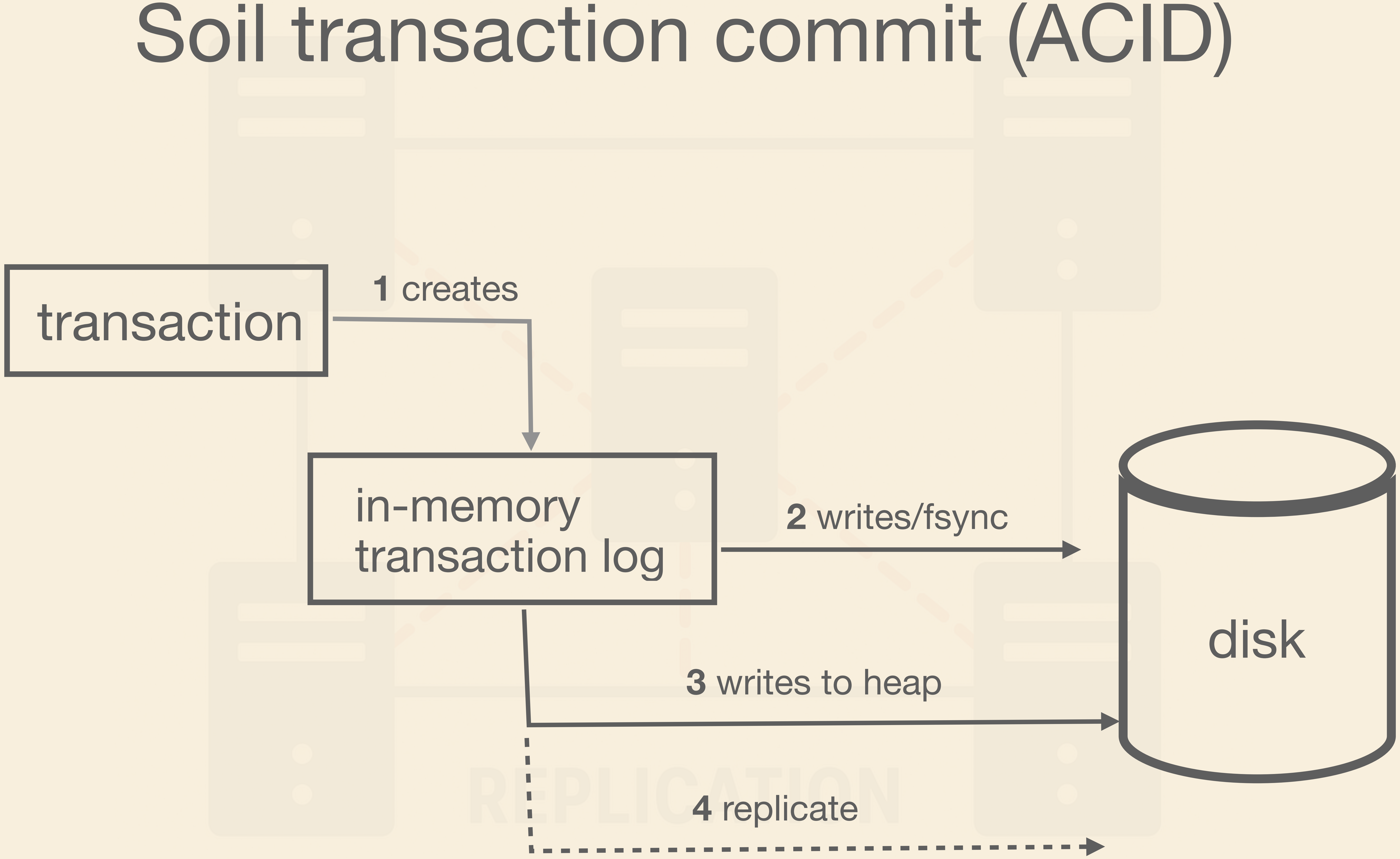


```
replicate: aTransactionLog

    url := '/logs' asZnUrl / aTransactionLog id asString.

    self nodes do: [ :node |
        self httpClient
            url: (url copy host: node ipAddress);
            entity: (ZnByteArrayEntity bytes: aTransactionLog soilSerialize);
            put
```

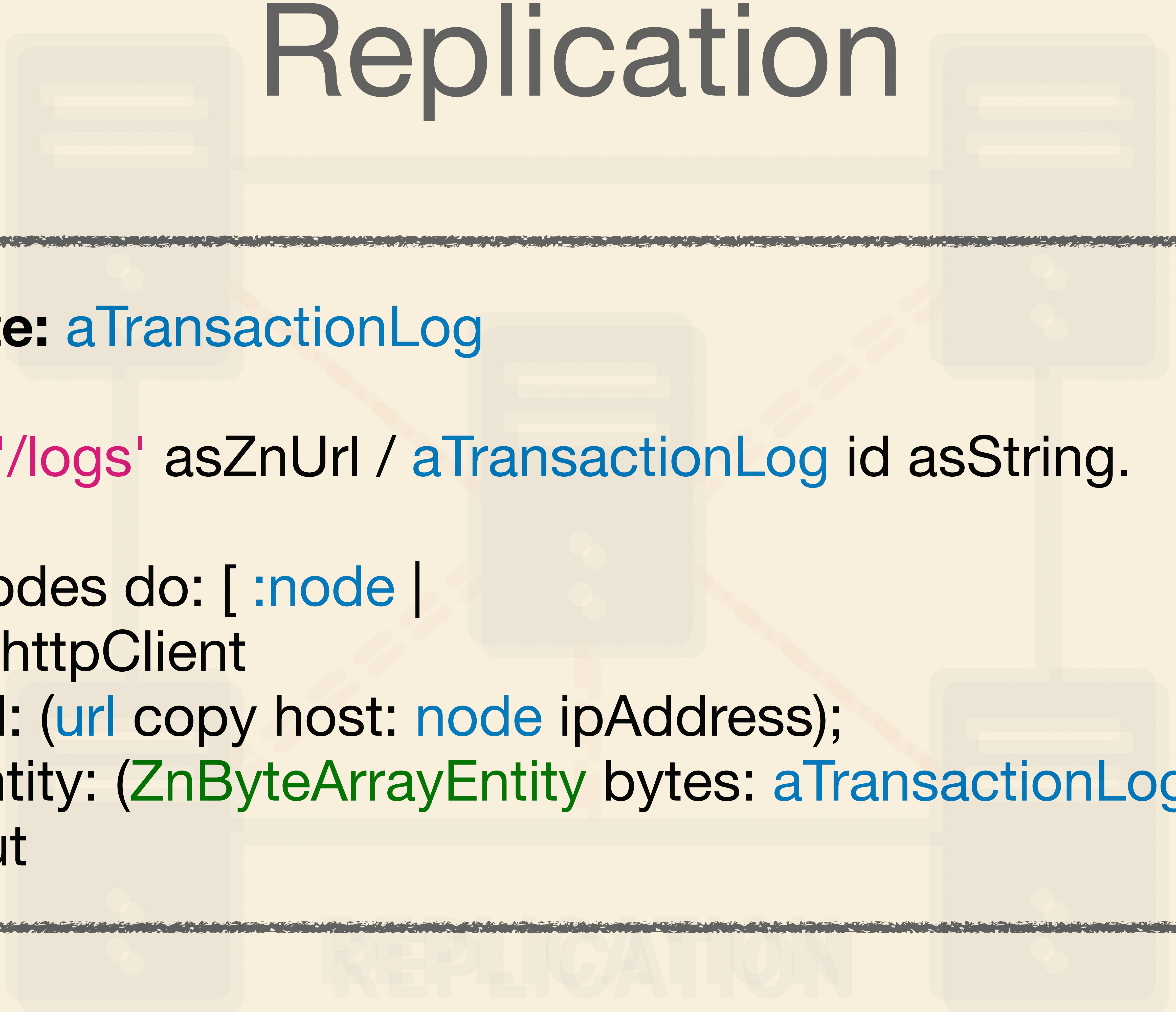
Soil transaction commit (ACID)



Soil transaction commit (ACID)



Replication



```
replicate: aTransactionLog

  url := '/logs' asZnUrl / aTransactionLog id asString.

  self nodes do: [ :node |
    self httpClient
      url: (url copy host: node ipAddress);
      entity: (ZnByteArrayEntity bytes: aTransactionLog soilSerialize);
      put
```

Replication

what happens if

$n > r$

Replication

what happens if

$$n > r$$

number of nodes (machines)

Replication

what happens if

$n > r$

number of
nodes (machines)

replication factor
(number of copies)

database discovery

which nodes have data for user id **1234**?

rendezvous hashing

build identifier node plus key to look up

id
node1:1234
node2:1234
node3:1234
node4:1234
node5:1234

rendezvous hashing

hash with any non-cryptographic hash algorithm

id	hash
node1:1234	46245C9D
node2:1234	26AEB165
node3:1234	499151F7
node4:1234	6D116468
node5:1234	A3938A63

rendezvous hashing

sort hash ascending (or descending)

id	hash
node2:1234	26AEB165
node1:1234	46245C9D
node3:1234	499151F7
node4:1234	6D116468
node5:1234	A3938A63

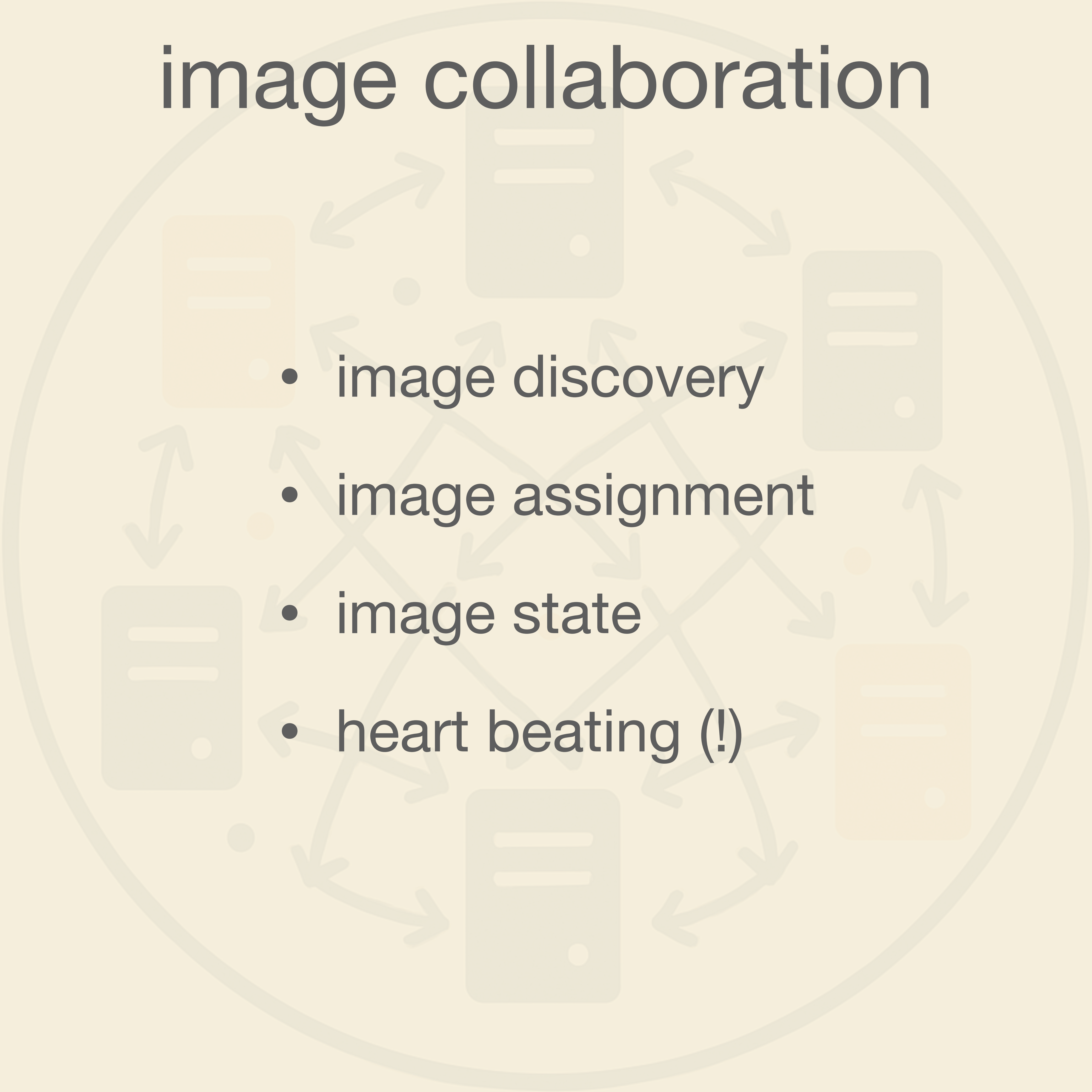
rendezvous hashing

take first <replication factor> items

r = 3

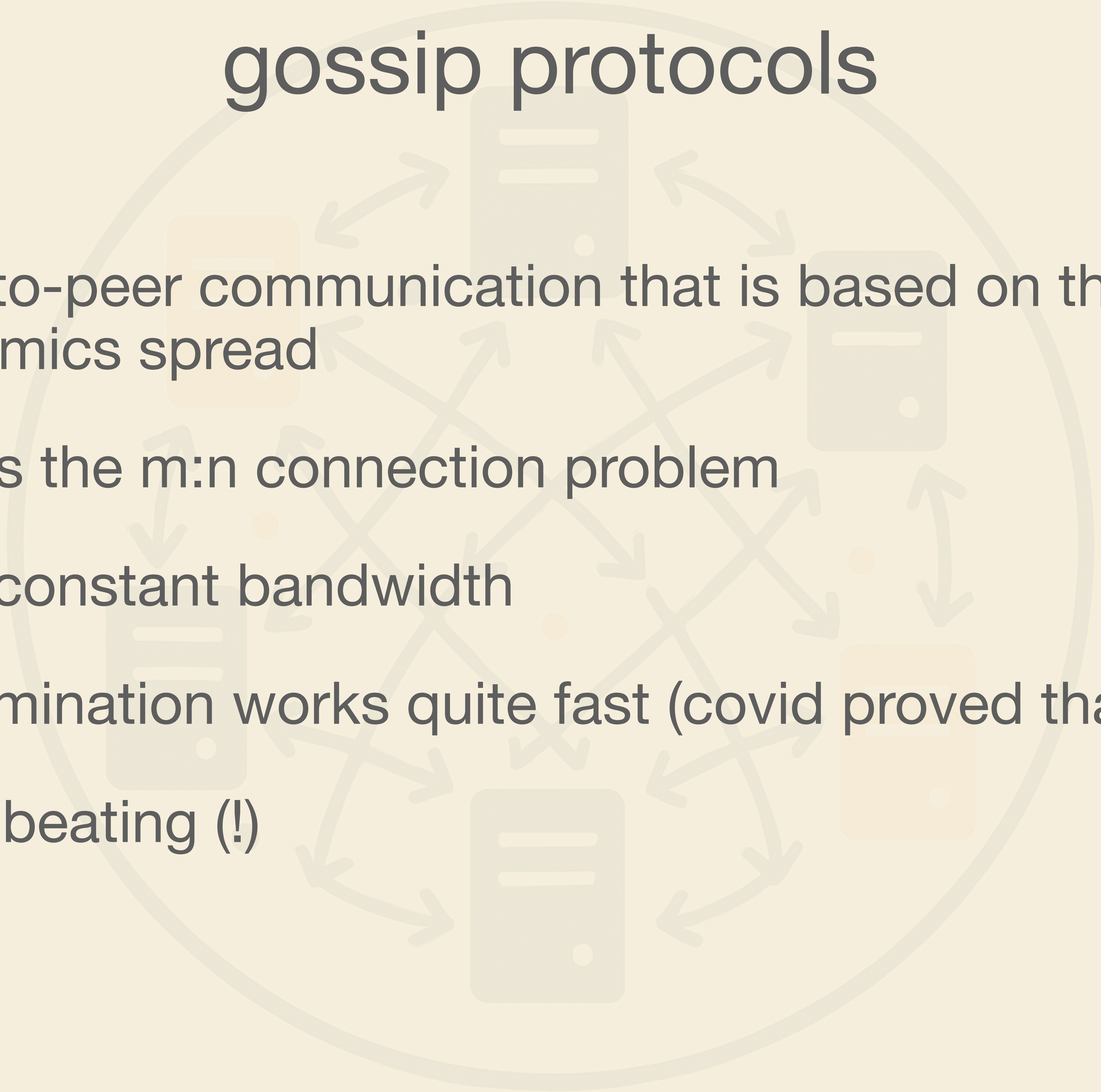
id	hash
node2:1234	26AEB165
node1:1234	46245C9D
node3:1234	499151F7
node4:1234	6D116468
node5:1234	A3938A63

image collaboration



- image discovery
- image assignment
- image state
- heart beating (!)

gossip protocols



- peer-to-peer communication that is based on the way epidemics spread
- avoids the m:n connection problem
- uses constant bandwidth
- dissemination works quite fast (covid proved that)
- heart beating (!)

demo

How to test all of this?

fallacies of distributed computing

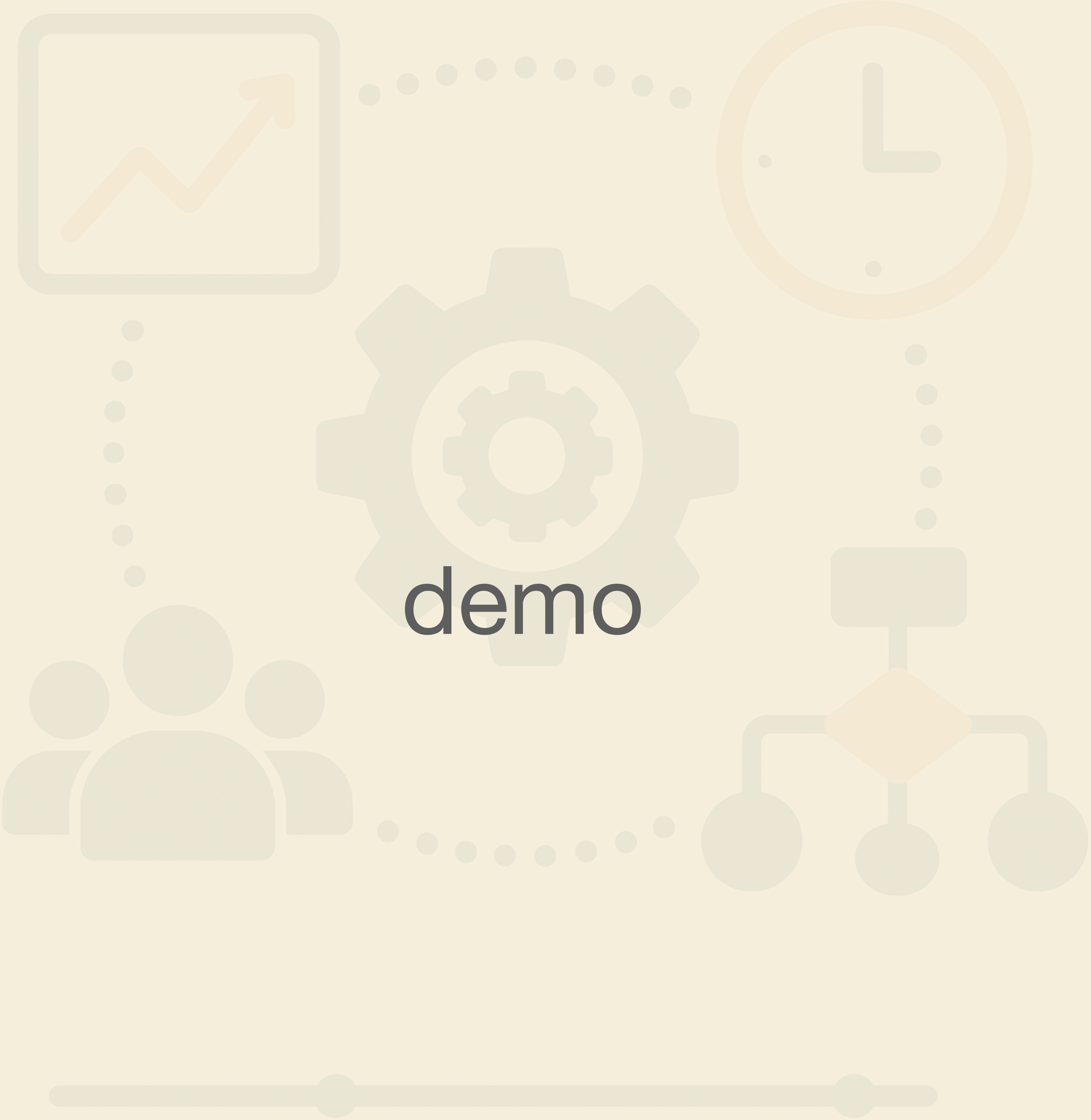
1. The network is reliable
2. Latency is zero
3. Bandwidth is infinite
4. The network is secure
5. Topology doesn't change
6. There is one administrator
7. Transport cost is zero
8. The network is homogeneous

DEVS

DISCRETE EVENT SIMULATION

Discrete Event System Specification

- port of SmallDEVS
- is a modular and hierarchical formalism for modeling and analyzing general systems
- is a timed event system



demo

thank you!

Keep your question for the end of the tutorial

ApptiveGrid

Norbert Hartl

norbert@apptivegrid.de