Soil

a database for fun and profit

Norbert Hartl - UK smalltalk user group - 26. Nov. 2025

I didn't choose the database life,
the database life chose me

#'Norbert Hartl' asSmalltalker

heard about smalltalk first in 1995

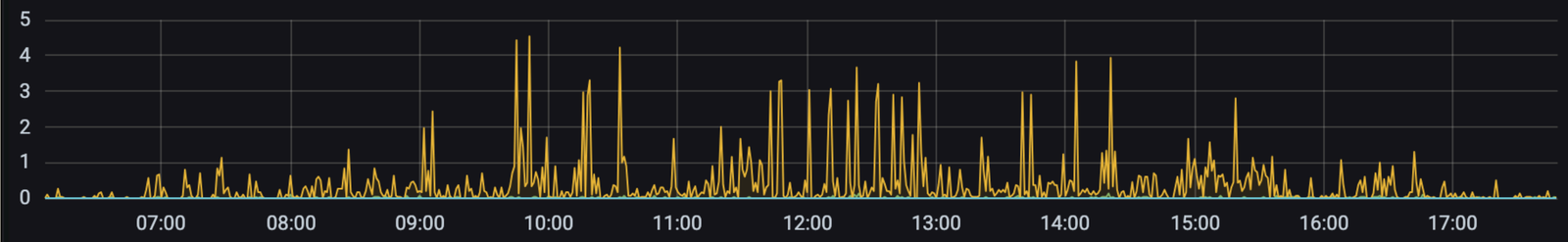
first professional project in 2005

= now 20 years of professional smalltalk

http://apptivegrid.de

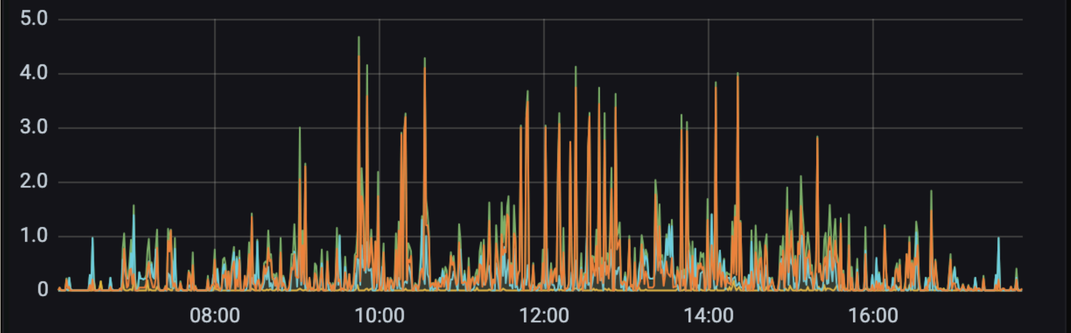
- a low-code/no-code tool for creating business processes
- a business IDE
- the reason to create soil

HTTP requests



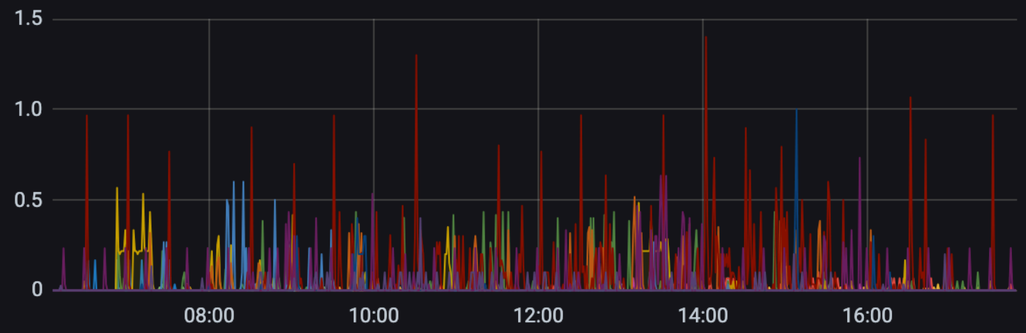
	max	avg	current
apptive-base	4.54	0.35	0.03
apptive-account	0.13	0.01	0

transactions



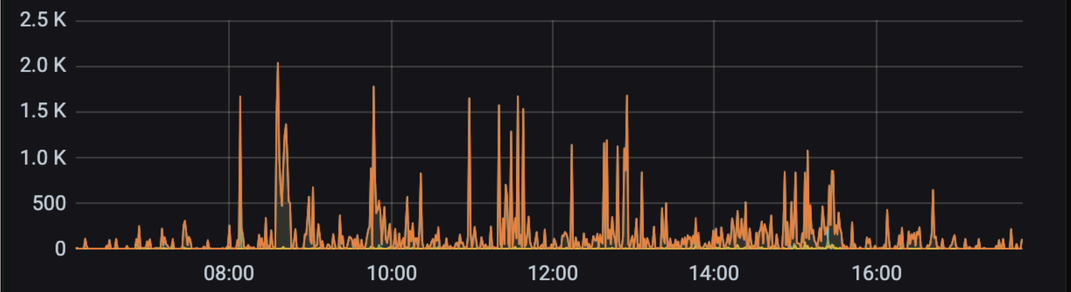
	max	avg	current
created	4.7	0.5	0.0
read-only	4.3	0.4	0.0
committed	1.4	0.2	0

transaction commits by instance



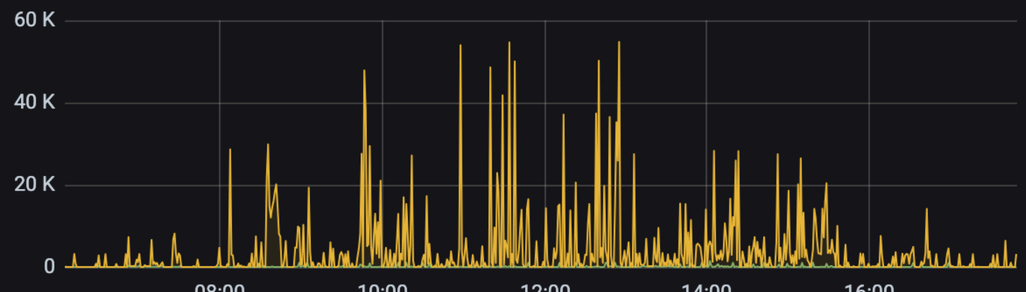
	max	avg	current
apptive2:3601 committed	1.4	0.1	0
apptive2:3603 committed	0.7	0.0	0
apptive1:3602 committed	0.6	0.0	0

cluster records



	max	avg	current
fetched	2.034 K	127	106
cache hits	2.033 K	125	106

object materializations



	max	avg	current
materialized	54.7 K	3.7 K	3.2 K
serialized	1.5 K	80	0

ApptiveGrid

It started ~ 1985 (40 years)

Got into SQL in 1995 (30 years)

mSQL

fun fact

The MySQL team originally tried to **license mSQL**, couldn't reach an agreement, and decided to create their **own engine** instead.

Struggles in 2005 (20 years)

GOODS, magma, ...

Got out of SQL in 2005 (20 years)

GOODS, magma, ... ended up with Glorp

...

Got out of SQL in 2005 (20 years)

GOODS, magma, ... ended up with Glorp
... then stopped using ORMs (and SQL)

From entities to documents in 2015 (10 years)

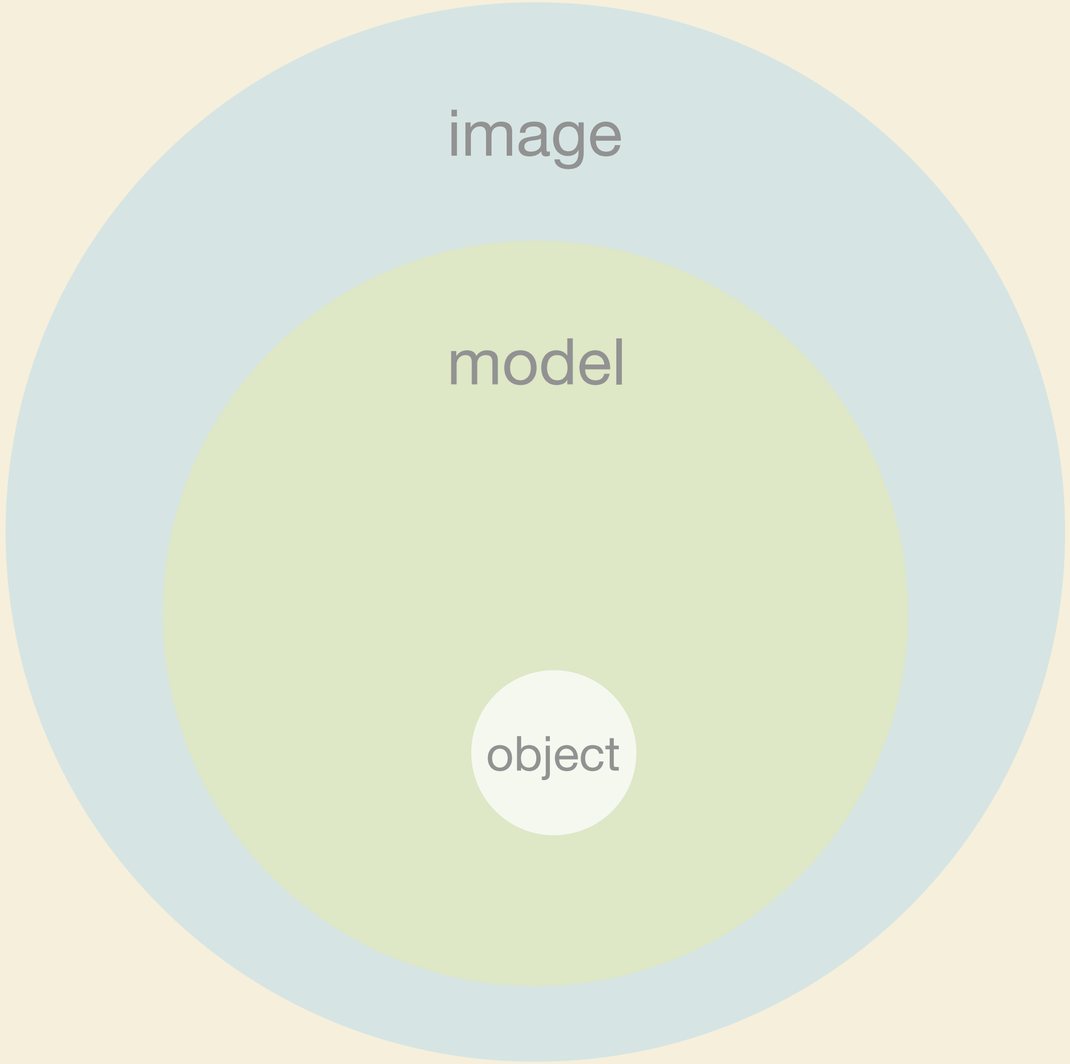
voyage + mongoDB

ApptiveGrid (5 years)

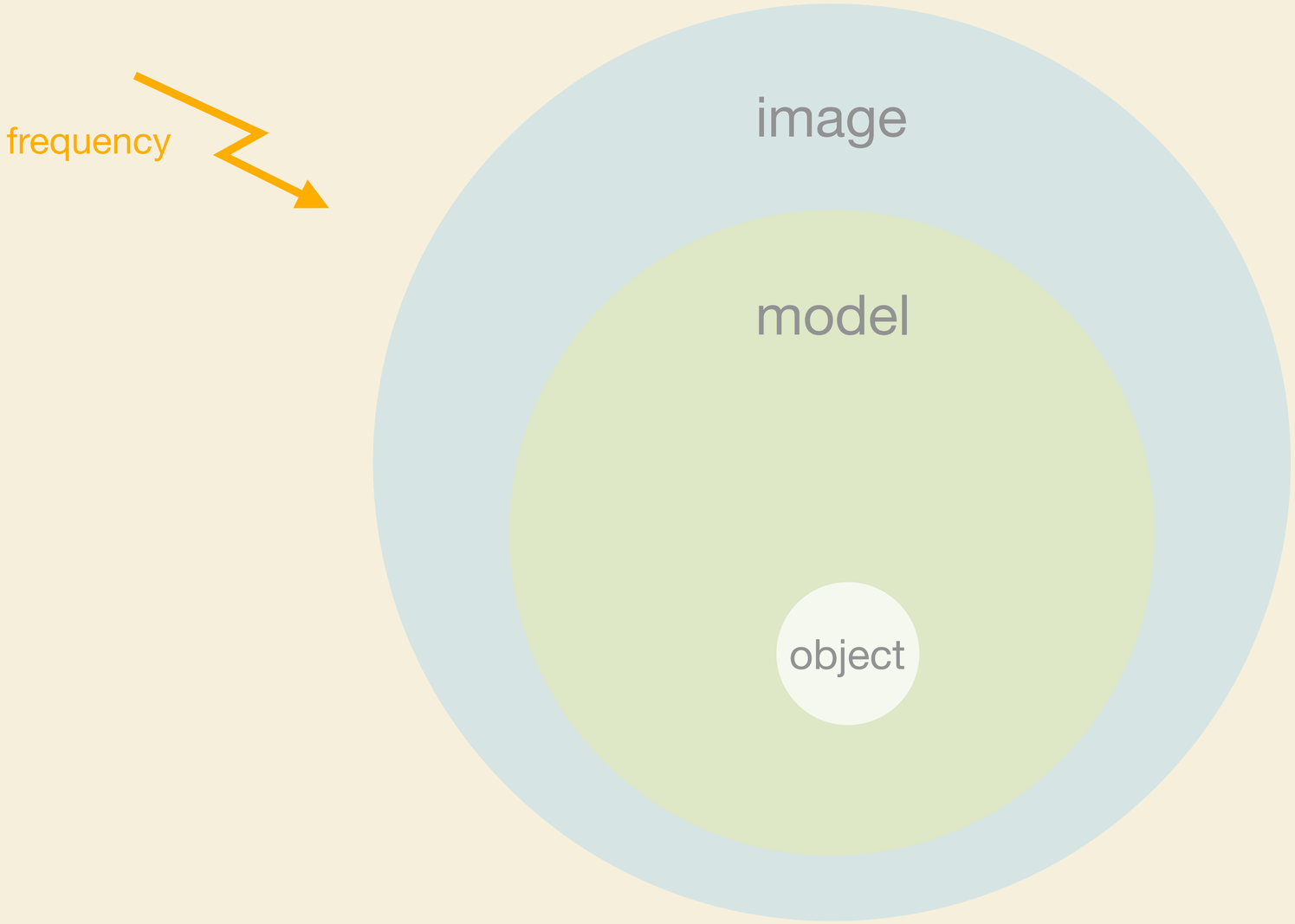
Q4 2020 - Q1 2021	image based persistence
Q2 2021 - Q1 2022	voyage + mongoDB
Q2 2022 - Q1 2023	OmniBase (work on soil started Q4 2022)
Q2 2023 -	soil

Persistence levels

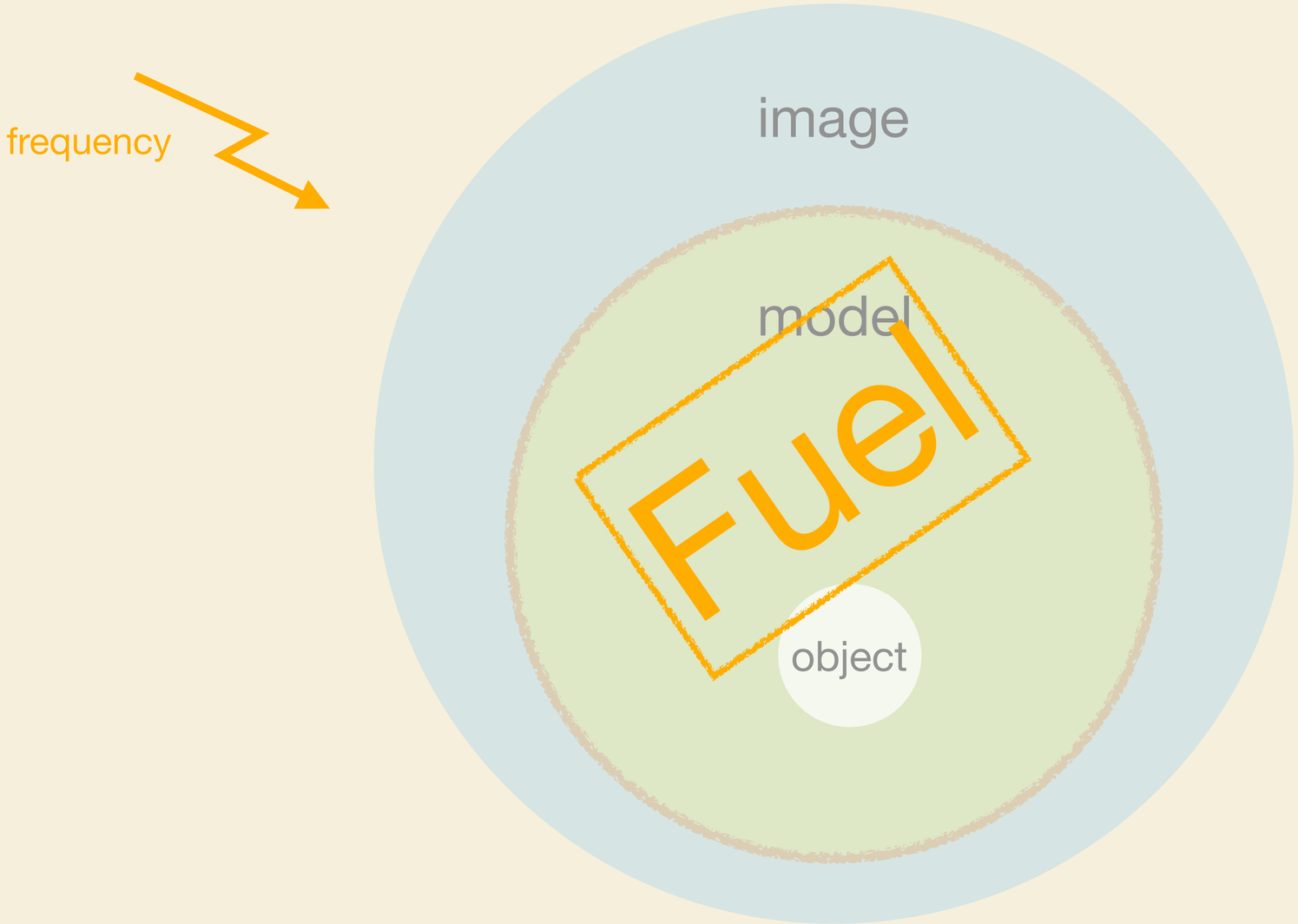
Persistence levels



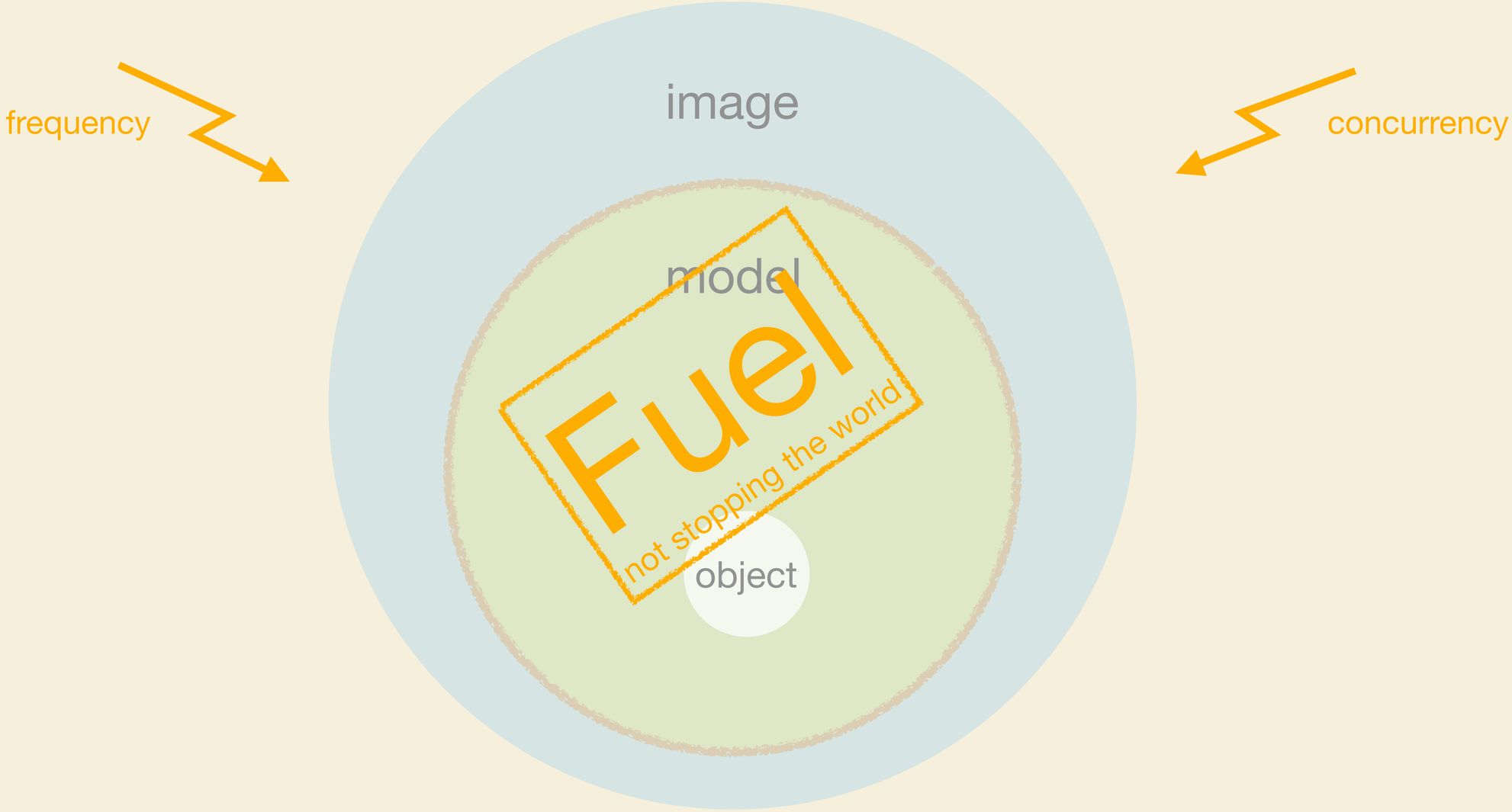
Persistence levels



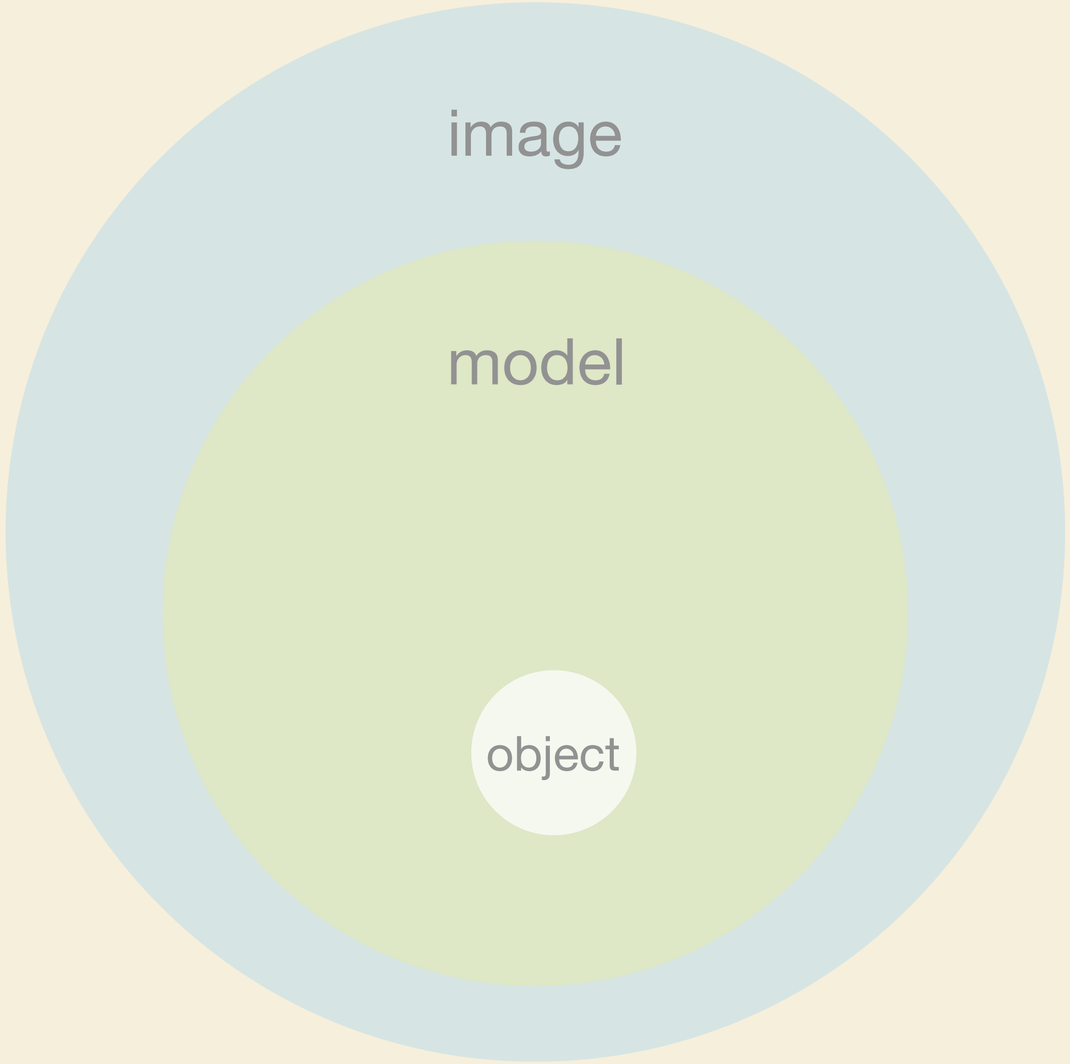
Persistence levels



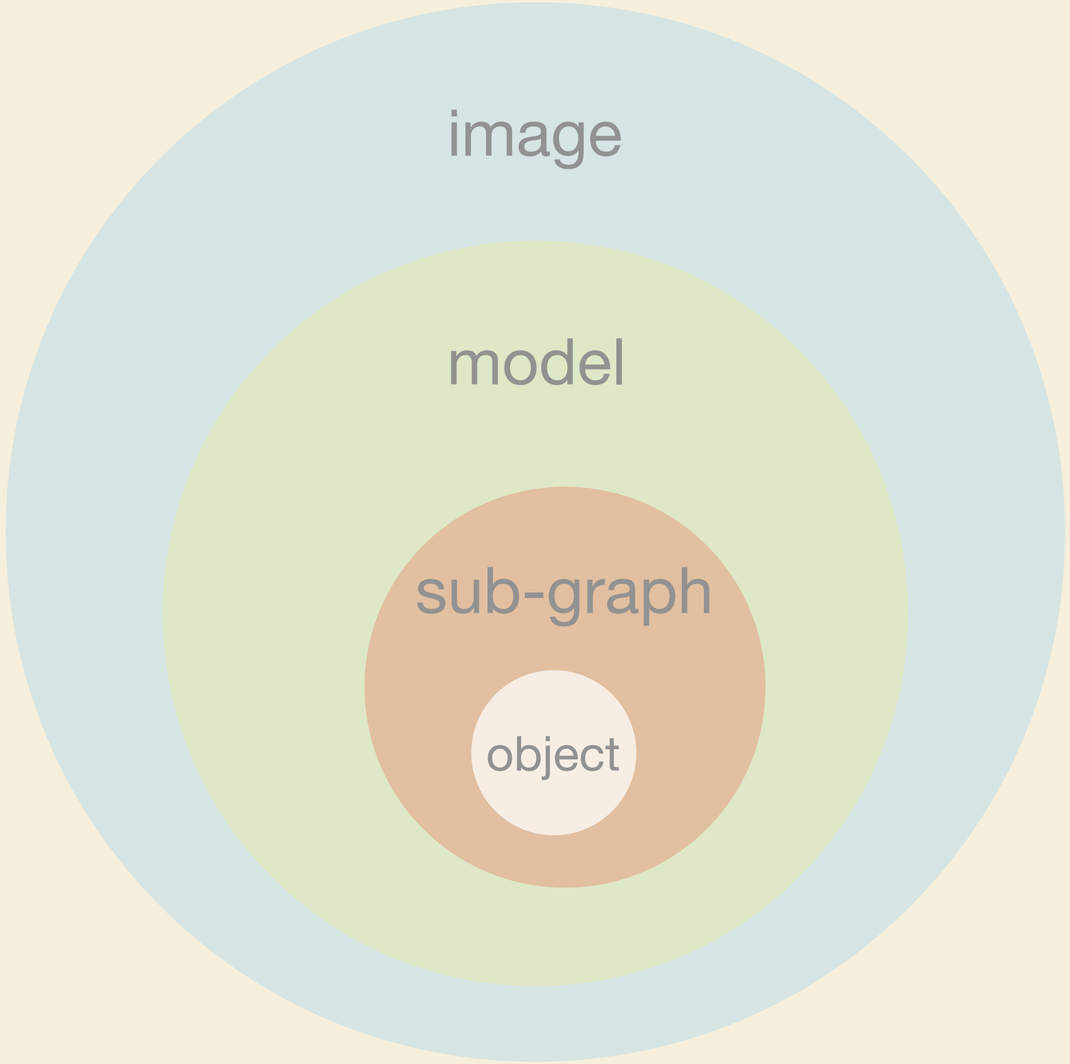
Persistence levels



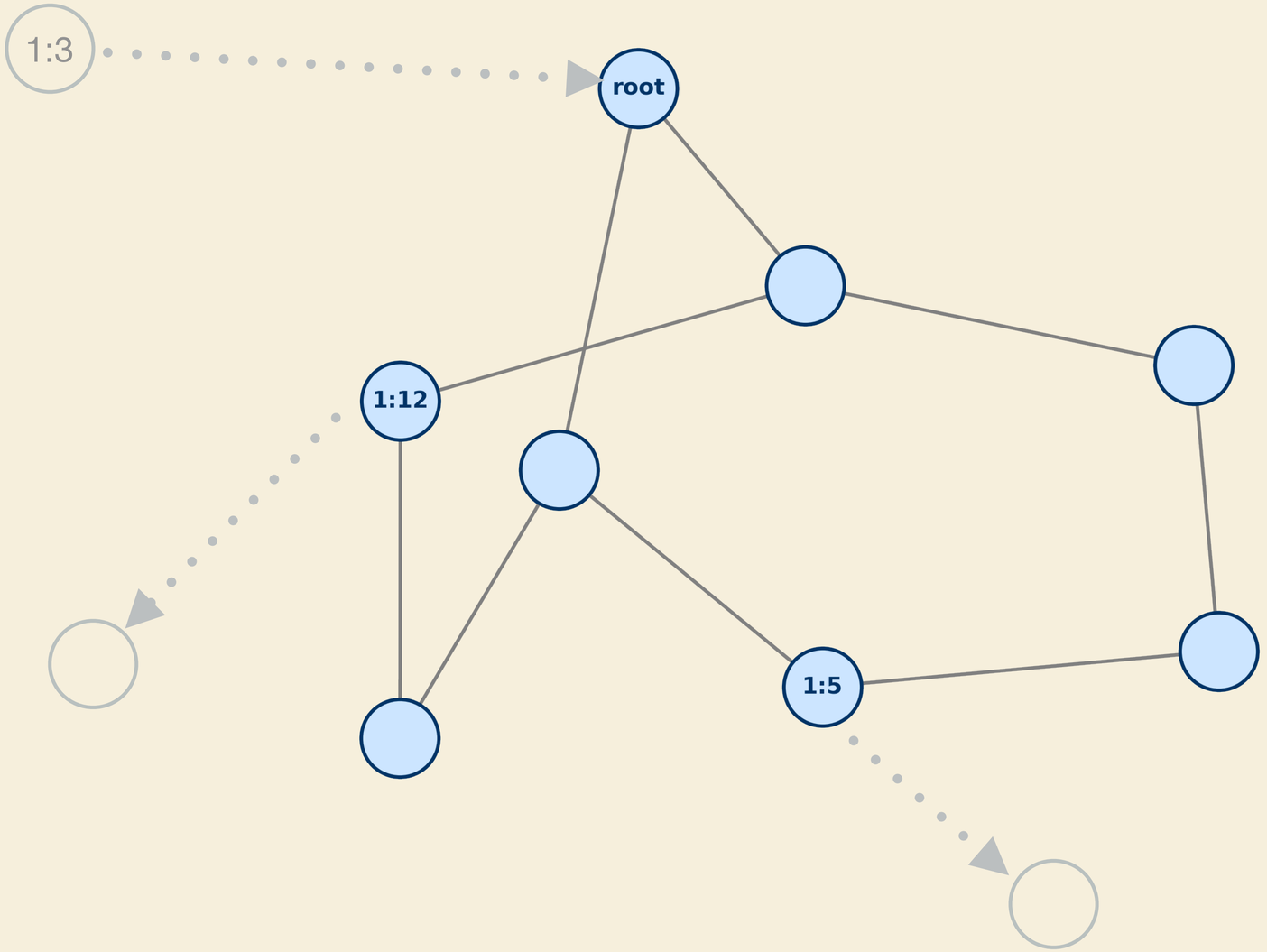
Persistence levels



Persistence levels



Sub graph = Cluster



Soil - Serializer

```
Object>>#soilBasicSerialize: serializer

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

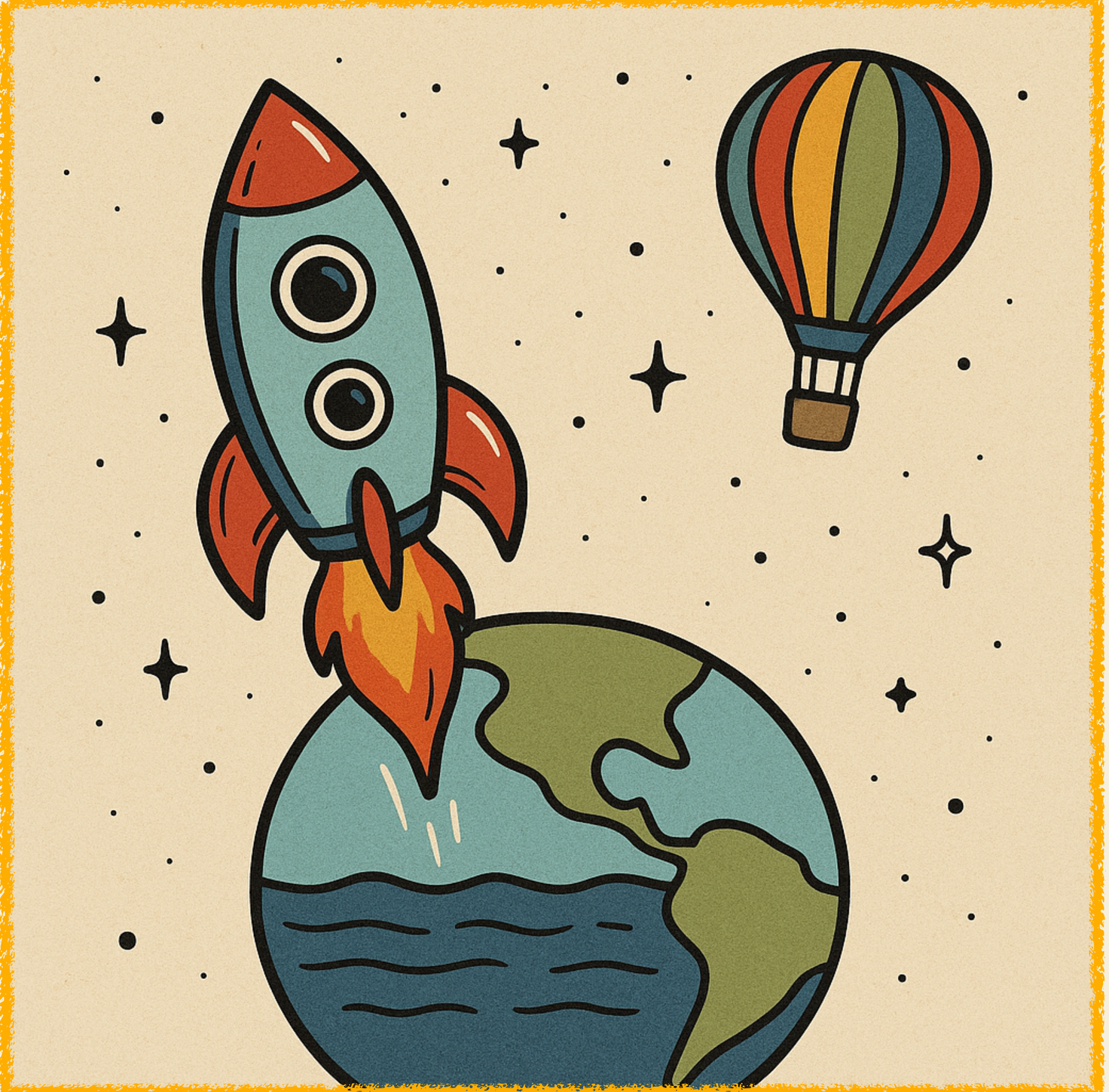
```
String>>#soilBasicSerialize: serializer

    serializer nextPutString: self
```

architecture bits

- one object copy per transaction
- MVCC (multi version concurrency control)
- ACID transactions
- Binary search indexes (skip list, BTree)
- Object segments

future bits

- indexes with duplicate key support
- secondary indexes
- lazy materialization
- indexes on the heap
- no markDirty anymore (object state)
- garbage collection
- distribution (Rhizome)
- ...



launch time

thank you!

ApptiveGrid

Norbert Hartl
norbert@apptivegrid.de