

Soil

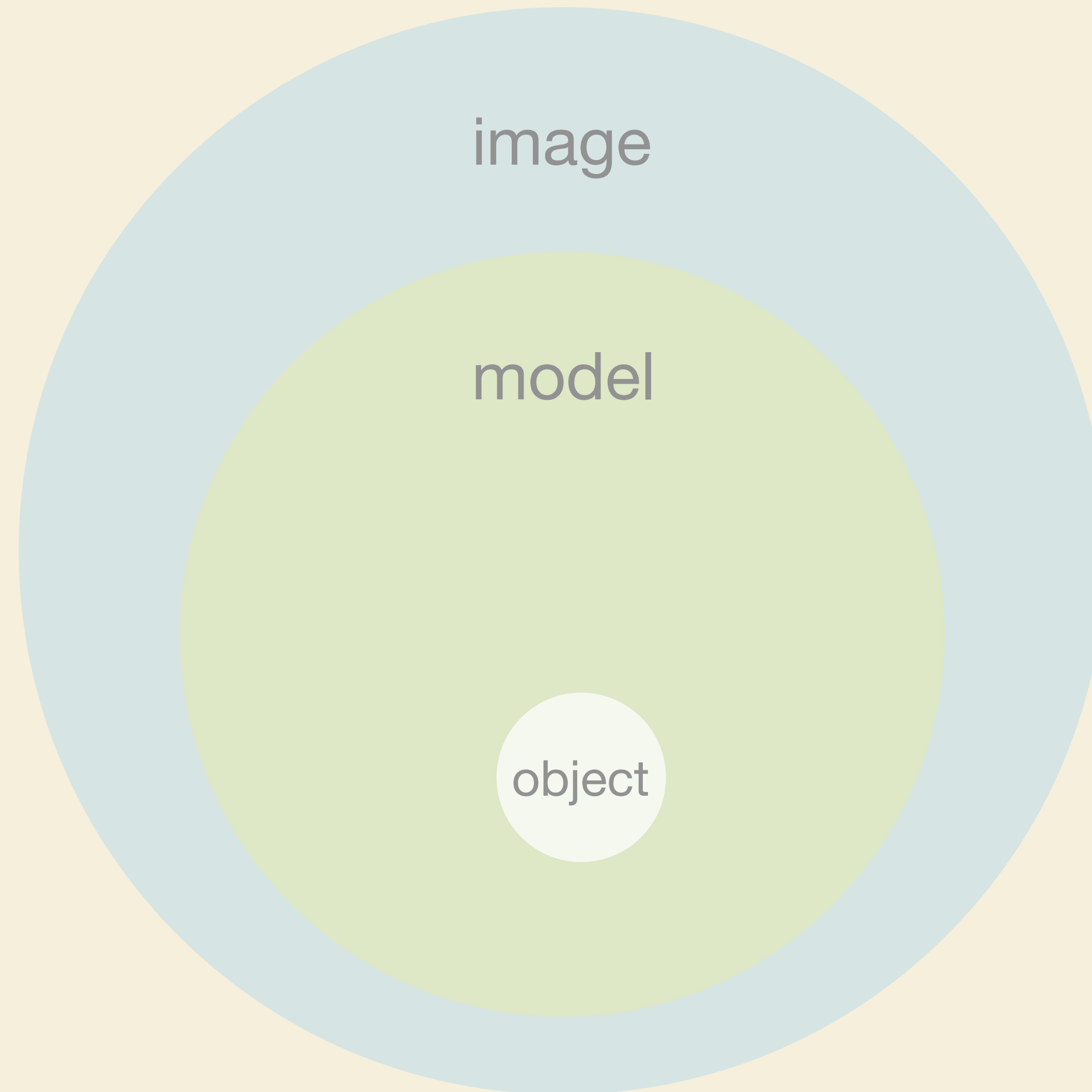
architectural insights

Soil - v5 released

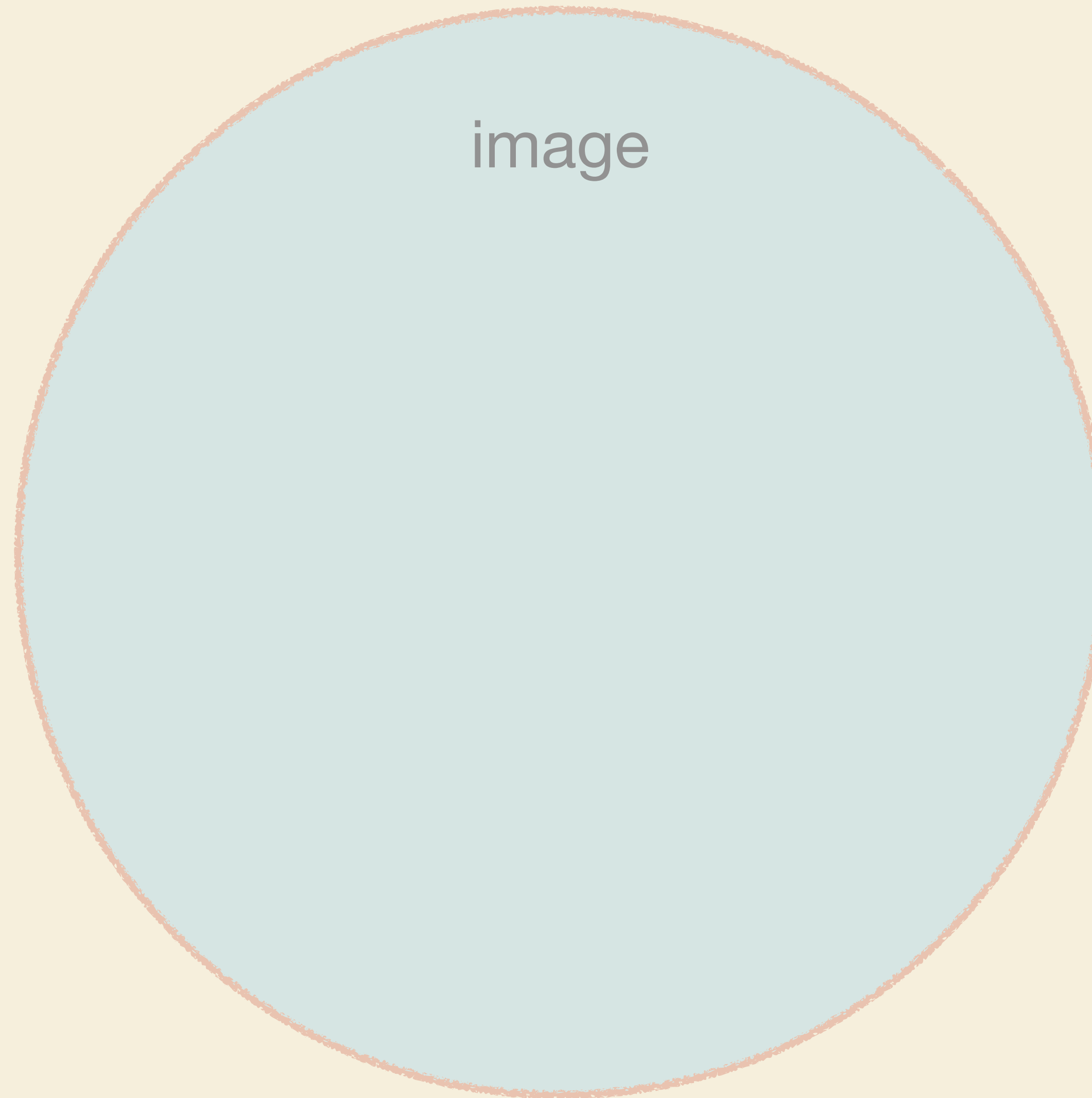
- new commit strategy
- automatic modification tracking
- no markDirty: anymore
- improved read-only transactions

what is the problem soil
is trying to solve?

known graph boundaries



Persistence levels - Image



Persistence levels - Image

- writes **all** objects (model, UI, ...)

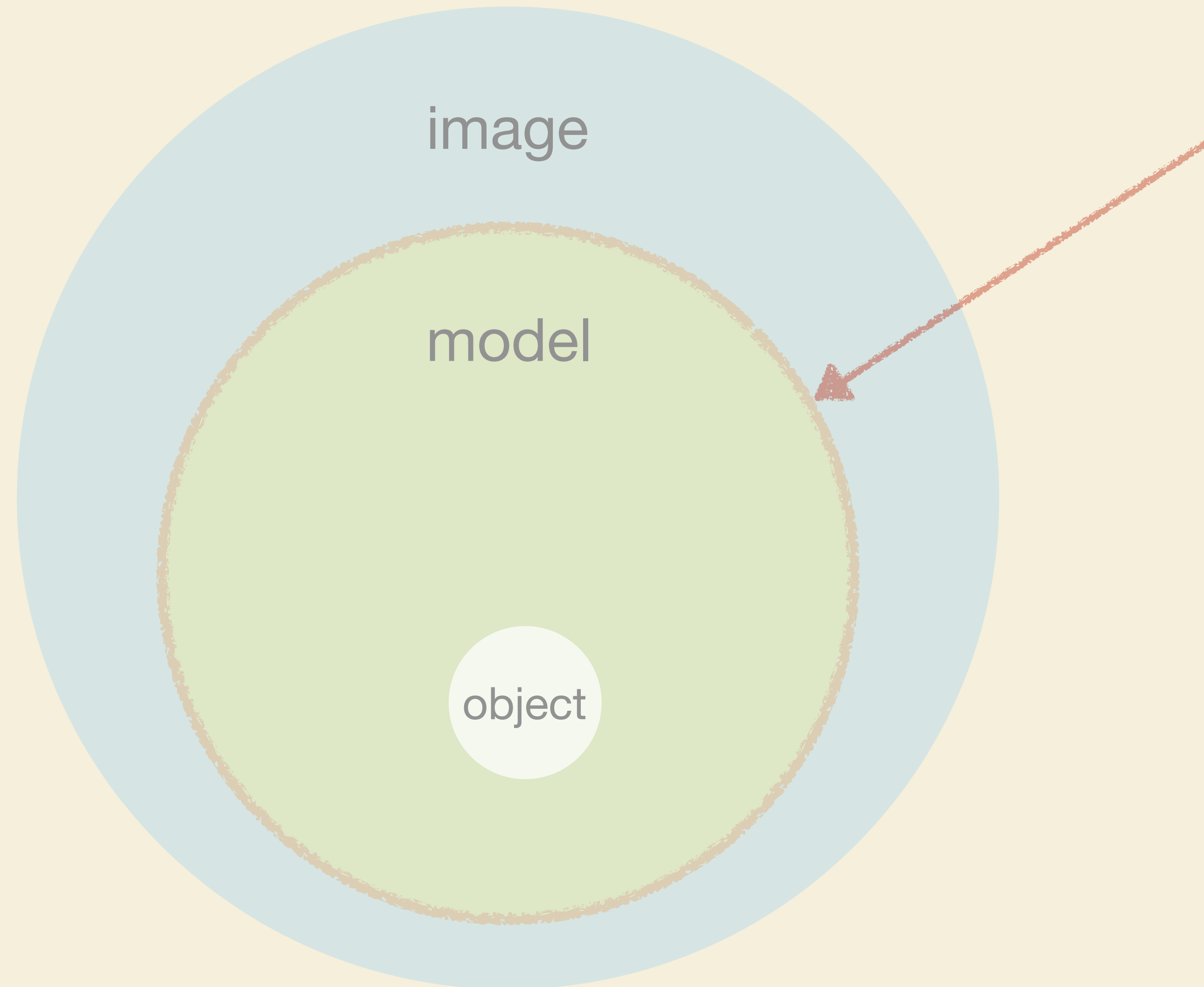
Persistence levels - Image

- writes **all** objects (model, UI, ...)
- stops the world

Persistence levels - Image

- writes **all** objects (model, UI, ...)
- stops the world
- limited to main memory

Persistence levels - Fuel



Persistence levels - Fuel

- writes only model (still big)

Persistence levels - Fuel

- writes only model (still big)
- does not stop the world

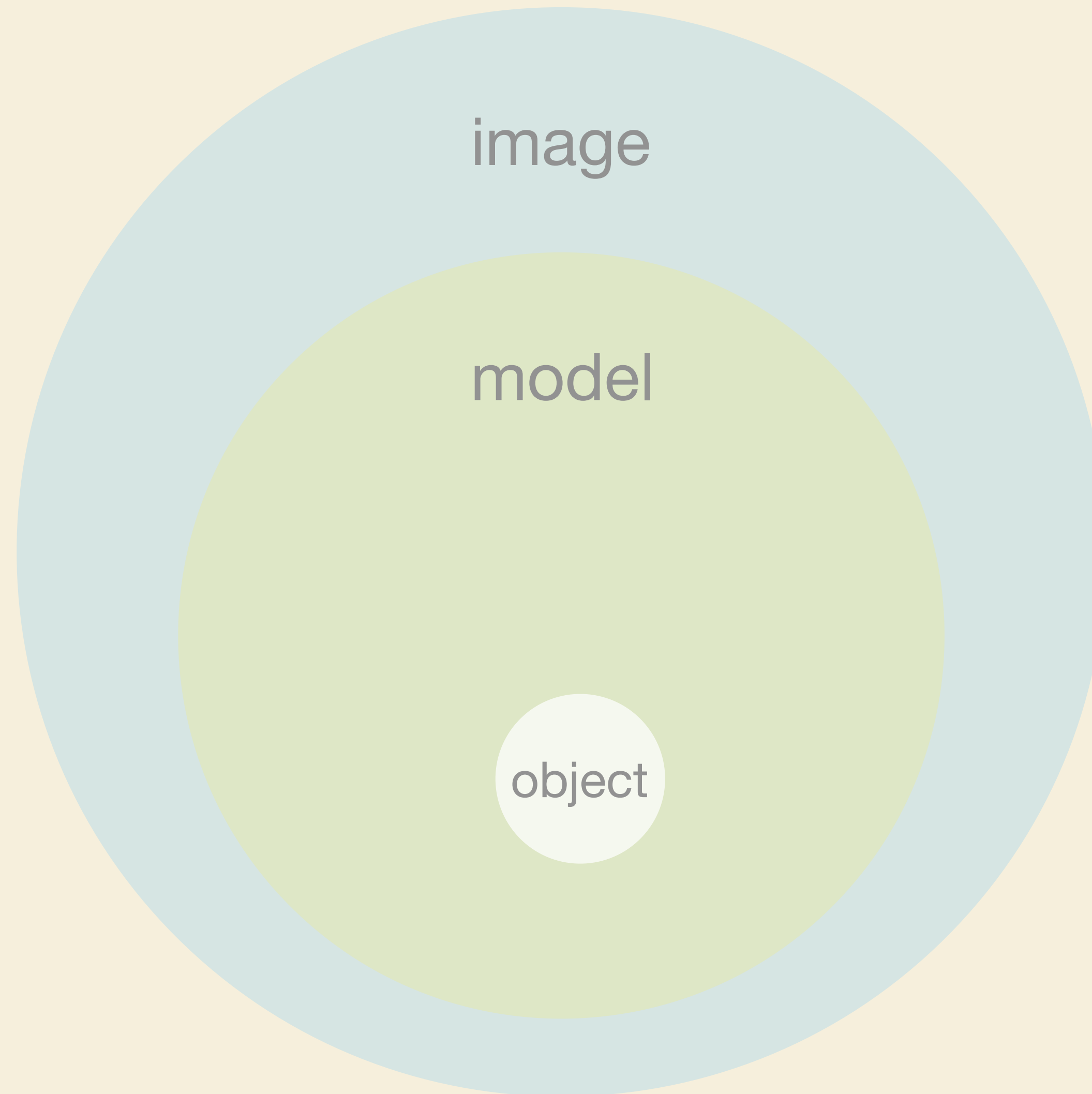
Persistence levels - Fuel

- writes only model (still big)
- does not stop the world
- limited to main memory

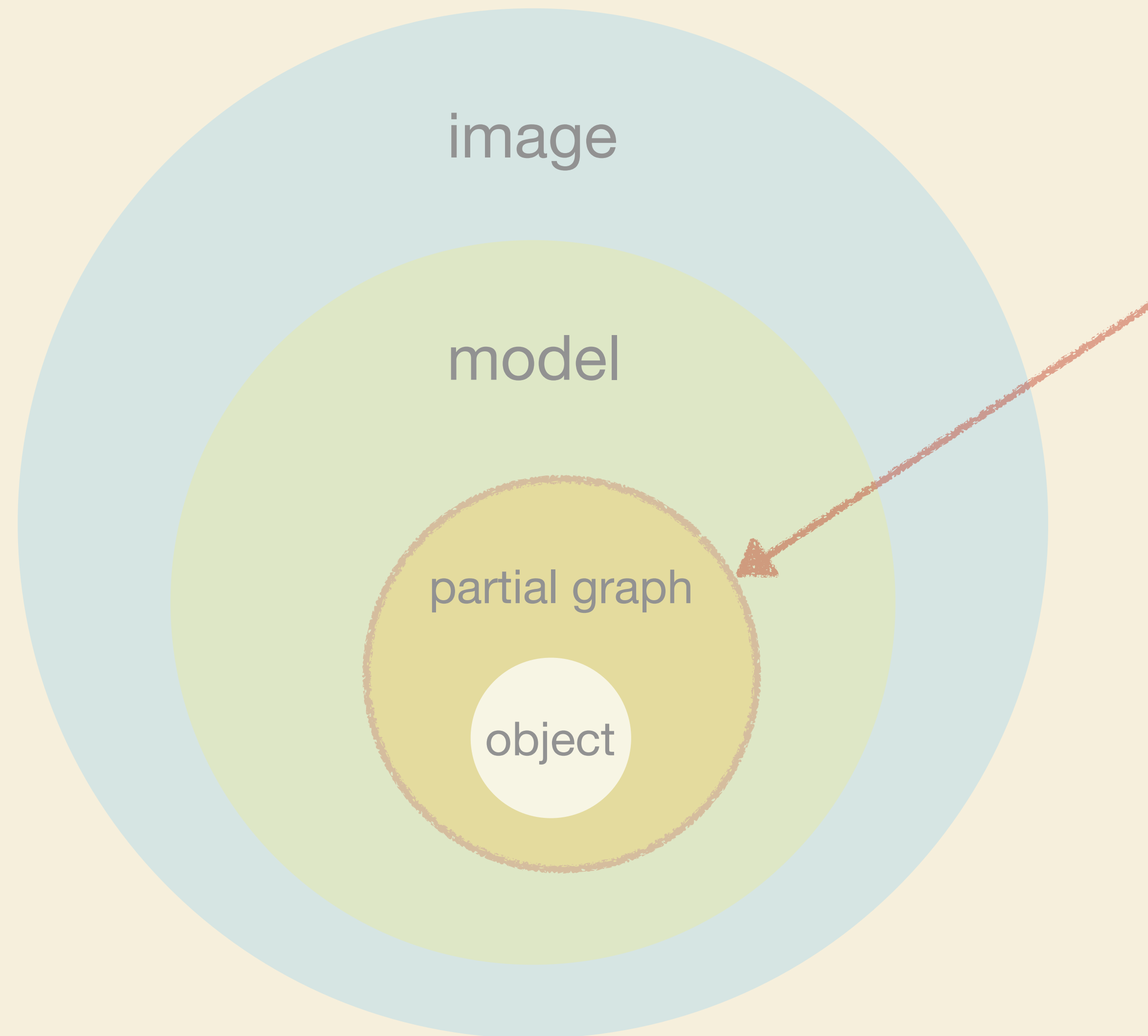
soil mission

write, read and query object graphs in an
efficient and concurrent safe way

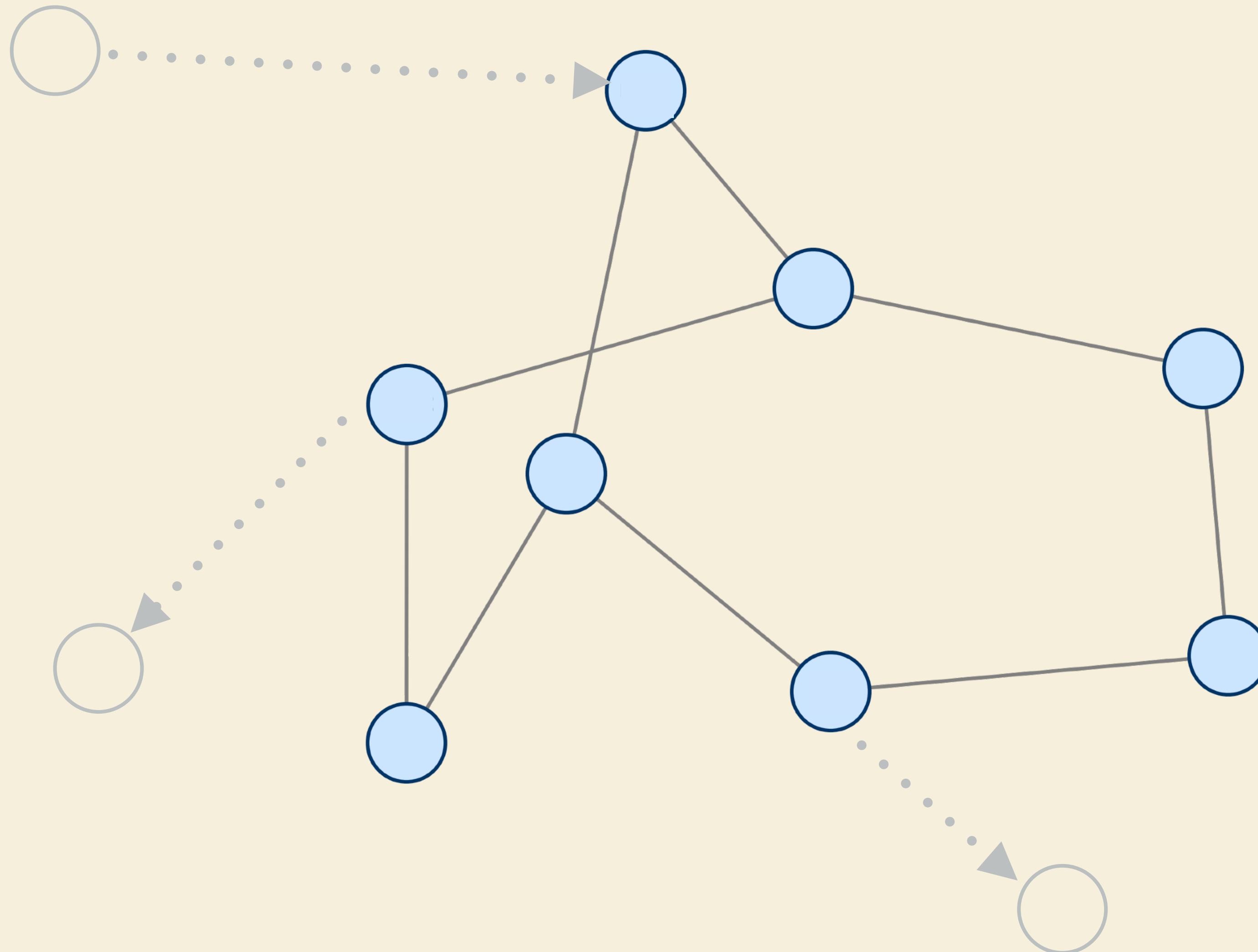
Persistence levels - Soil



Persistence levels - Soil



partial graph



partial graph

- small size allows fast reads/writes

partial graph

- small size allows fast reads/writes
- non-shared copy in transaction enables concurrency

partial graph

- small size allows fast reads/writes
- non-shared copy in transaction enables concurrency
- enable models to be bigger than main memory

partial graph - root objects

```
...  
transaction makeRoot: anObject  
...
```

- graph granularity is application-specific

partial graph - root objects



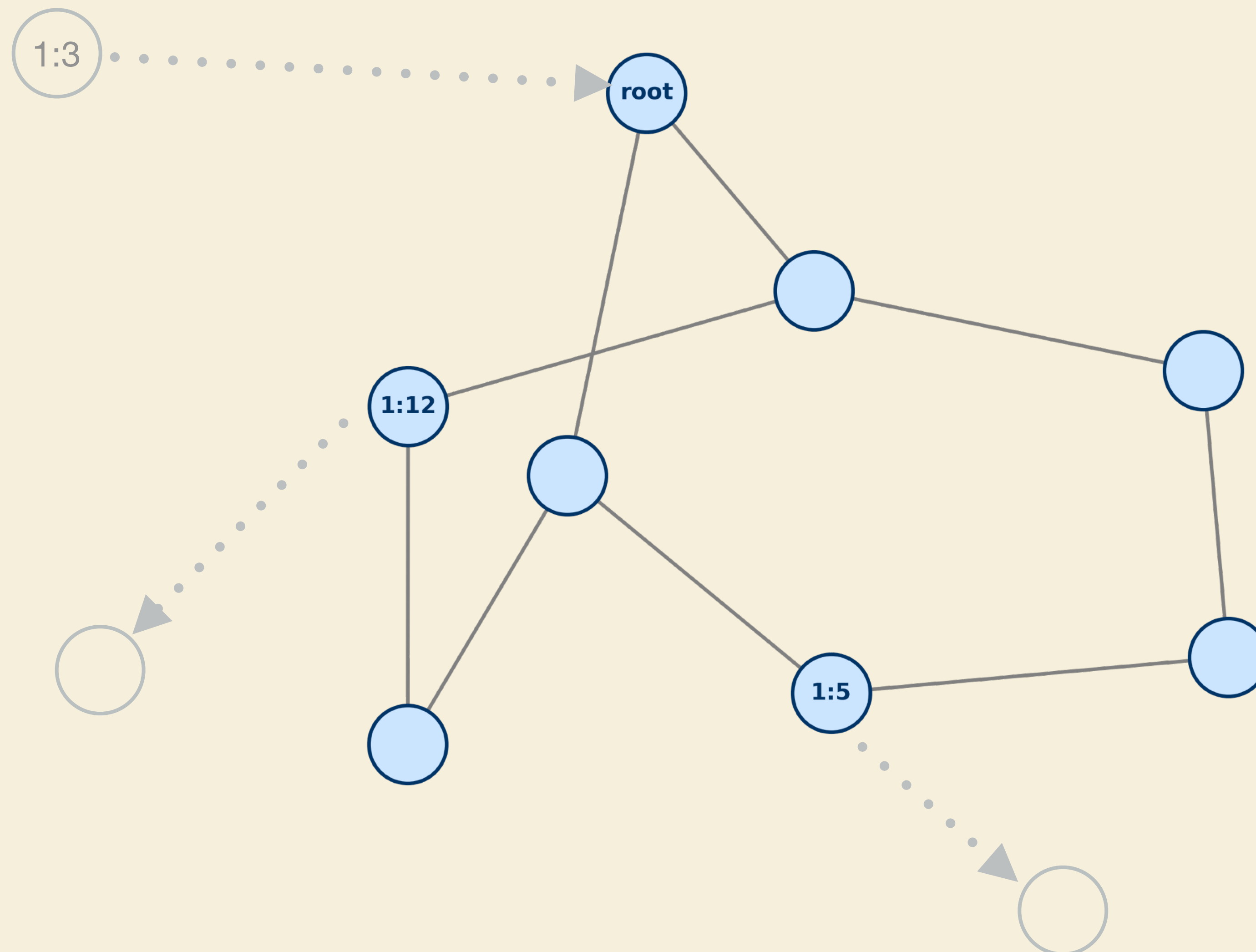
- graph granularity is application-specific
- root object gets ObjectId as address

partial graph - root objects

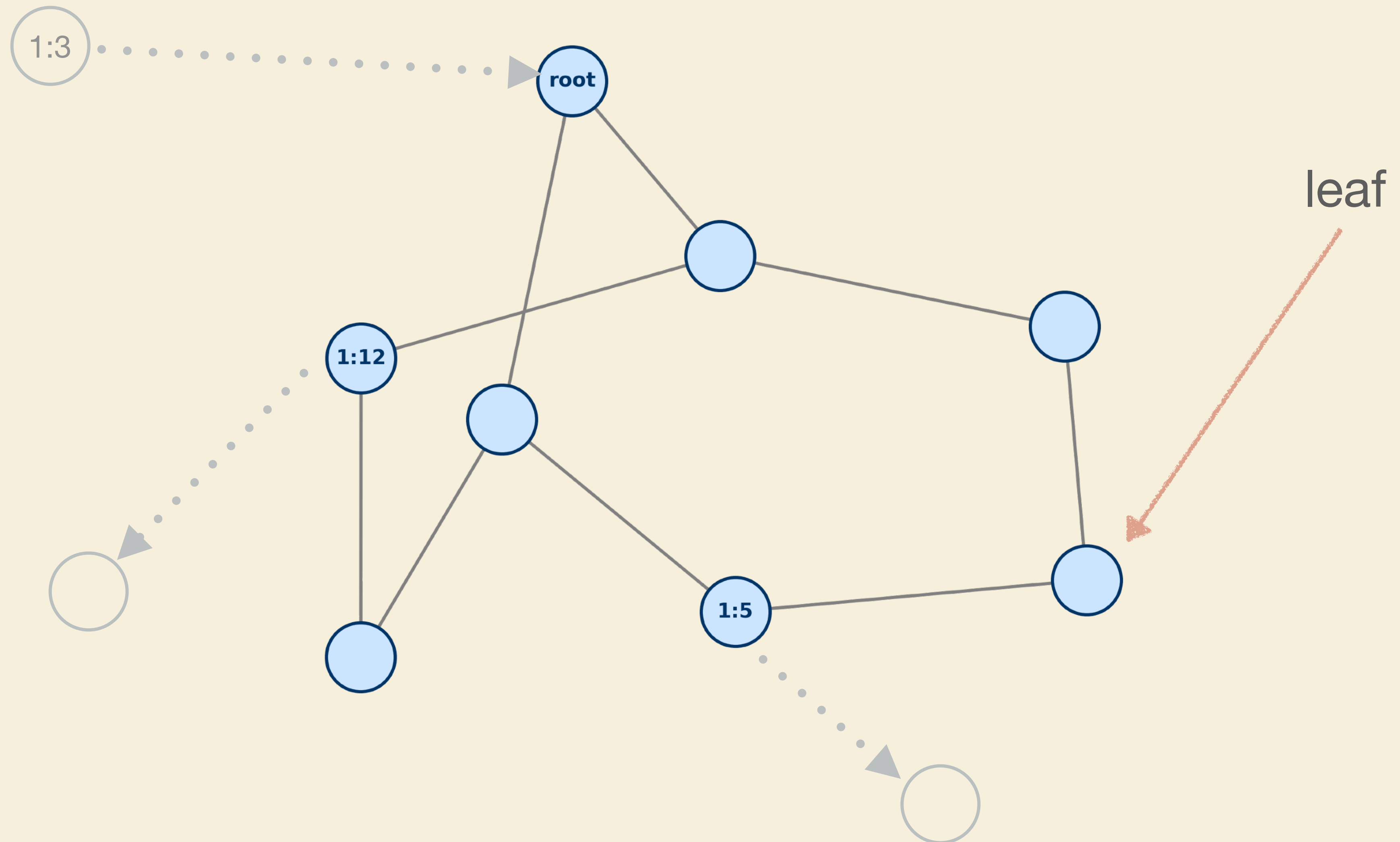


- graph granularity is application-specific
- root object gets ObjectId as address
- defines traversal boundaries...

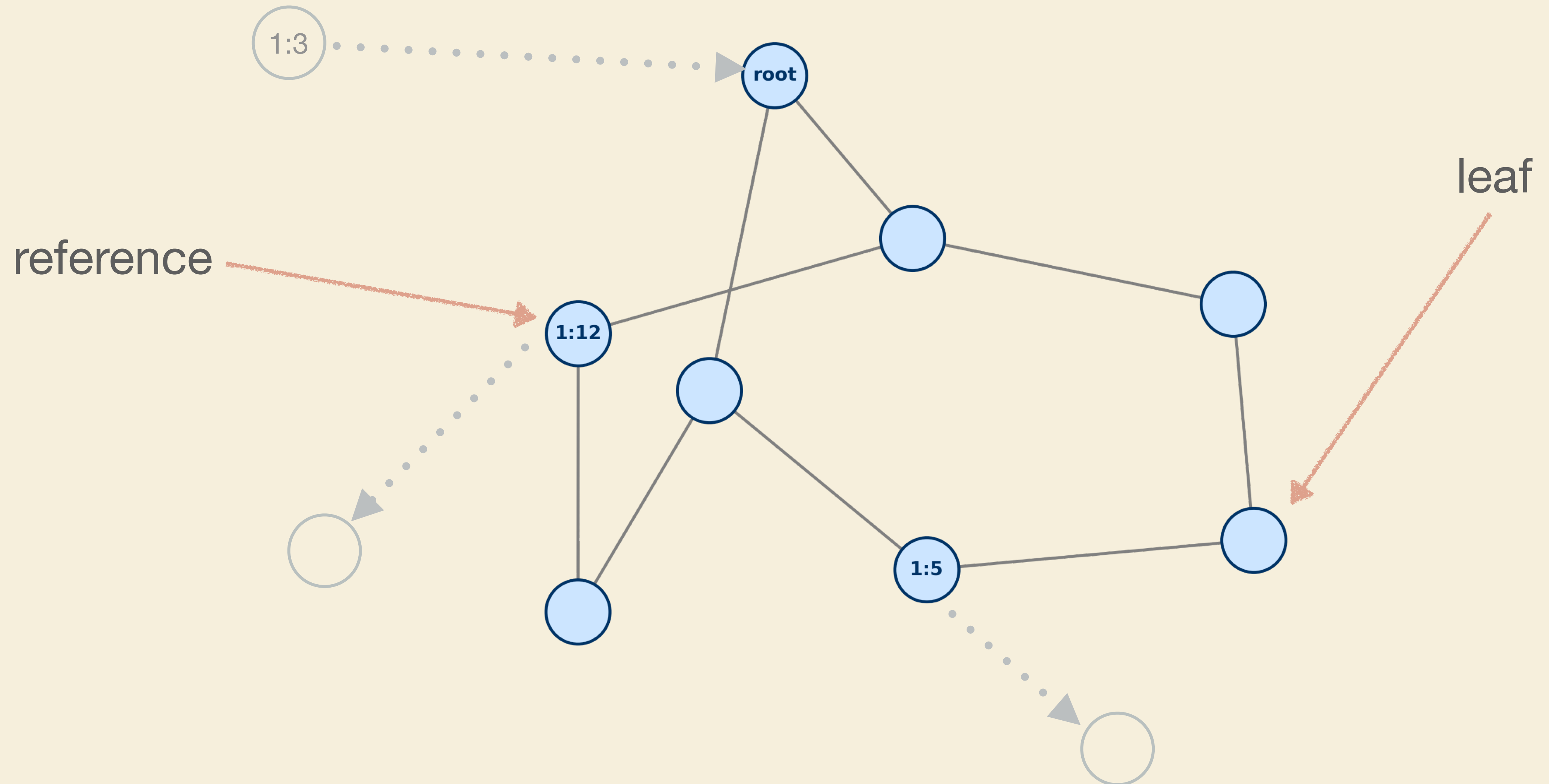
partial graph / cluster



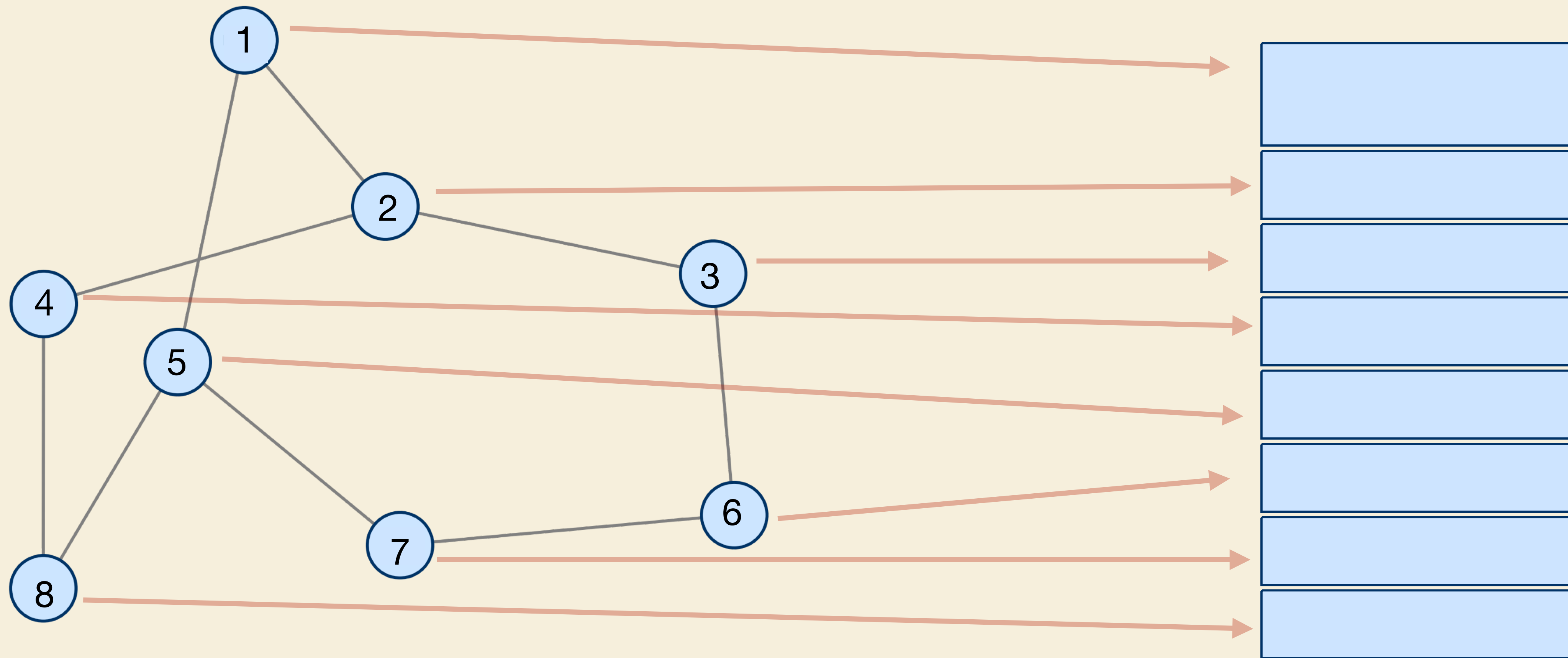
partial graph / cluster



partial graph / cluster



serialization - traverse and write



serialize an object

```
Object>>#soilBasicSerialize: serializer
```

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

"normal" objects

serialize an object

```
Object>>#soilBasicSerialize: serializer
```

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

"normal" objects

```
Integer>>#soilBasicSerialize: serializer
```

```
serializer nextInteger: self
```

immediate objects

serialize an object

```
Object>>#soilBasicSerialize: serializer
```

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

"normal" objects

```
Integer>>#soilBasicSerialize: serializer
```

```
serializer nextInteger: self
```

immediate objects

```
String>>#soilBasicSerialize: serializer
```

```
serializer nextPutString: self
```

custom serialization

serialize an object

**deterministic
traversal**

```
Object>>#soilBasicSerialize: serializer
```

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

"normal" objects

```
Integer>>#soilBasicSerialize: serializer
```

```
serializer nextInteger: self
```

immediate objects

```
String>>#soilBasicSerialize: serializer
```

```
serializer nextPutString: self
```

custom serialization

storage format

type code	r	e	c	o	r	d	b	y	t	e	s
--------------	---	---	---	---	---	---	---	---	---	---	---

initializeTypeCodes

```
<script>
```

```
TypeCodeObject := 1.
```

```
TypeCodeUndefined := 2.
```

```
TypeCodeTrue := 3.
```

```
TypeCodeFalse := 4.
```

```
TypeCodePositiveInteger := 5.
```

```
TypeCodeNegativeInteger := 6.
```

```
TypeCodeFloat := 7.
```

```
TypeCodeFraction := 8.
```

```
TypeCodeScaledDecimal := 9.
```

```
TypeCodeCharacter := 10.
```

behavior description

- object stored in meta segment

behavior description

- object stored in meta segment
- stores instance variable names and order

behavior description

- object stored in meta segment
- stores instance variable names and order
- has version

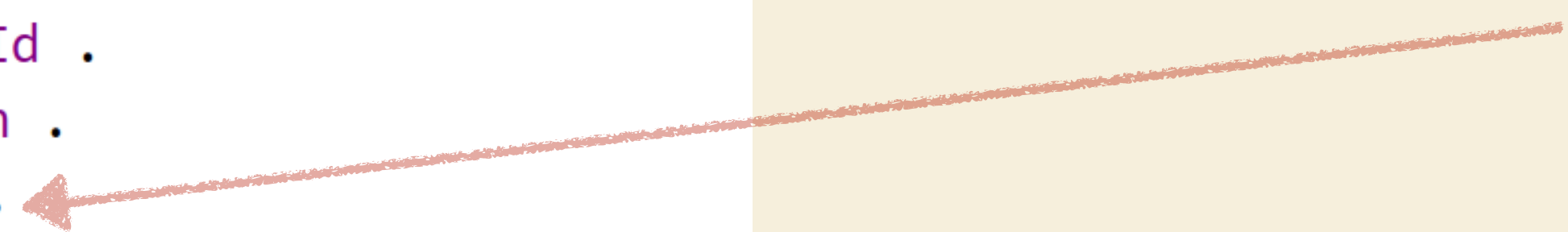
behavior description

- object stored in meta segment
- stores instance variable names and order
- has version
- migrates old objects to current class

cluster record

```
Object << #SoilClusterRecord
  slots: {
    #objectId .
    #version .
    #bytes .
    #behaviorDescriptions .
    #references .
    #previousVersionPosition .
  };
tag: 'Model';
package: 'Soil-Core'
```

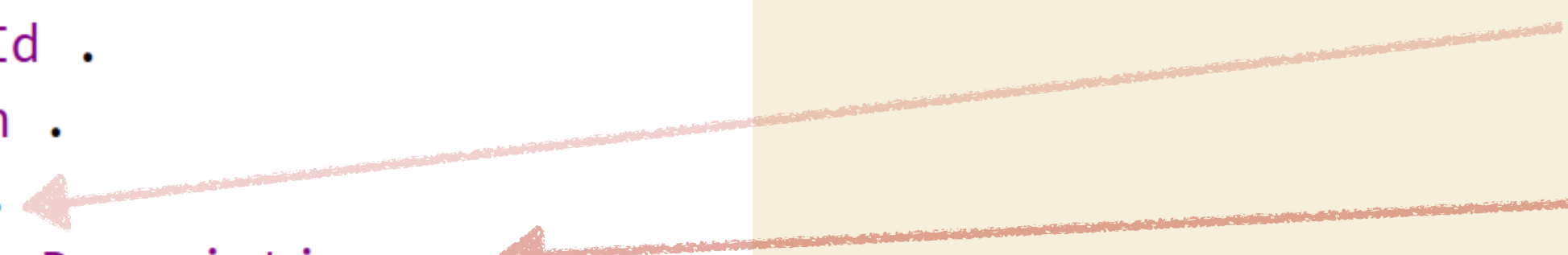
serialized partial graph



cluster record

```
Object << #SoilClusterRecord
  slots: {
    #objectId .
    #version .
    #bytes .
    #behaviorDescriptions .
    #references .
    #previousVersionPosition .
  };
  tag: 'Model';
  package: 'Soil-Core'
```

list of versioned ObjectIds (1:3:2)



cluster record

```
Object << #SoilClusterRecord
```

```
  slots: {
```

```
    #objectId .
```

```
    #version .
```

```
    #bytes .
```

```
    #behaviorDescriptions .
```

```
    #references .
```

```
    #previousVersionPosition .
```

```
  };
```

```
  tag: 'Model';
```

```
  package: 'Soil-Core'
```

list of ObjectIds (1:3)



cluster record

```
Object << #SoilClusterRecord
  slots: {
    #objectId .
    #version .
    #bytes .
    #behaviorDescriptions .
    #references .
    #previousVersionPosition .
  };
tag: 'Model';
package: 'Soil-Core'
```

version of record (MVCC)

position of previous version

cluster record

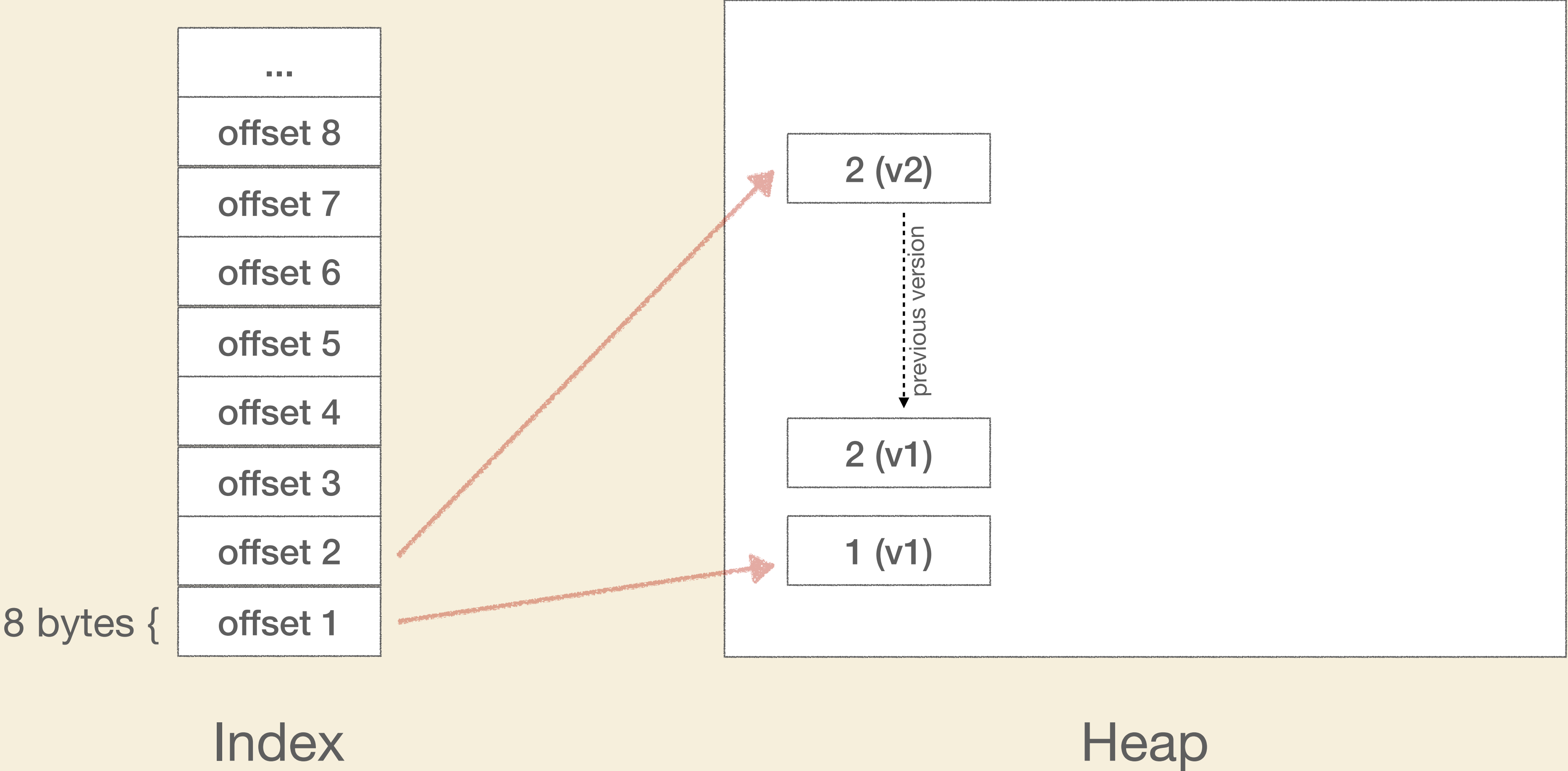
```
Object << #SoilClusterRecord
  slots: {
    #objectId .
    #version .
    #bytes .
    #behaviorDescriptions .
    #references .
    #previousVersionPosition .
  };
  tag: 'Model';
  package: 'Soil-Core'
```

version of record (MVCC)

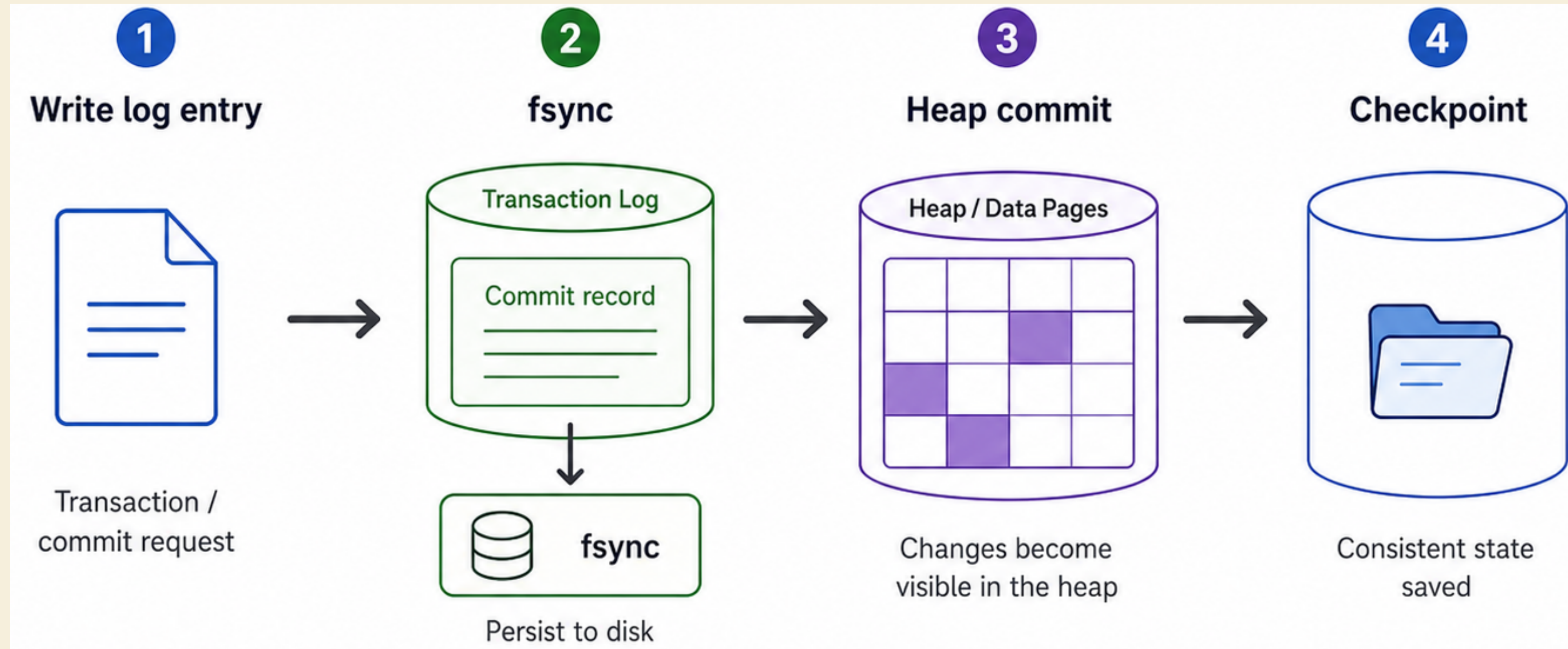
position of previous version

**application model
independent**

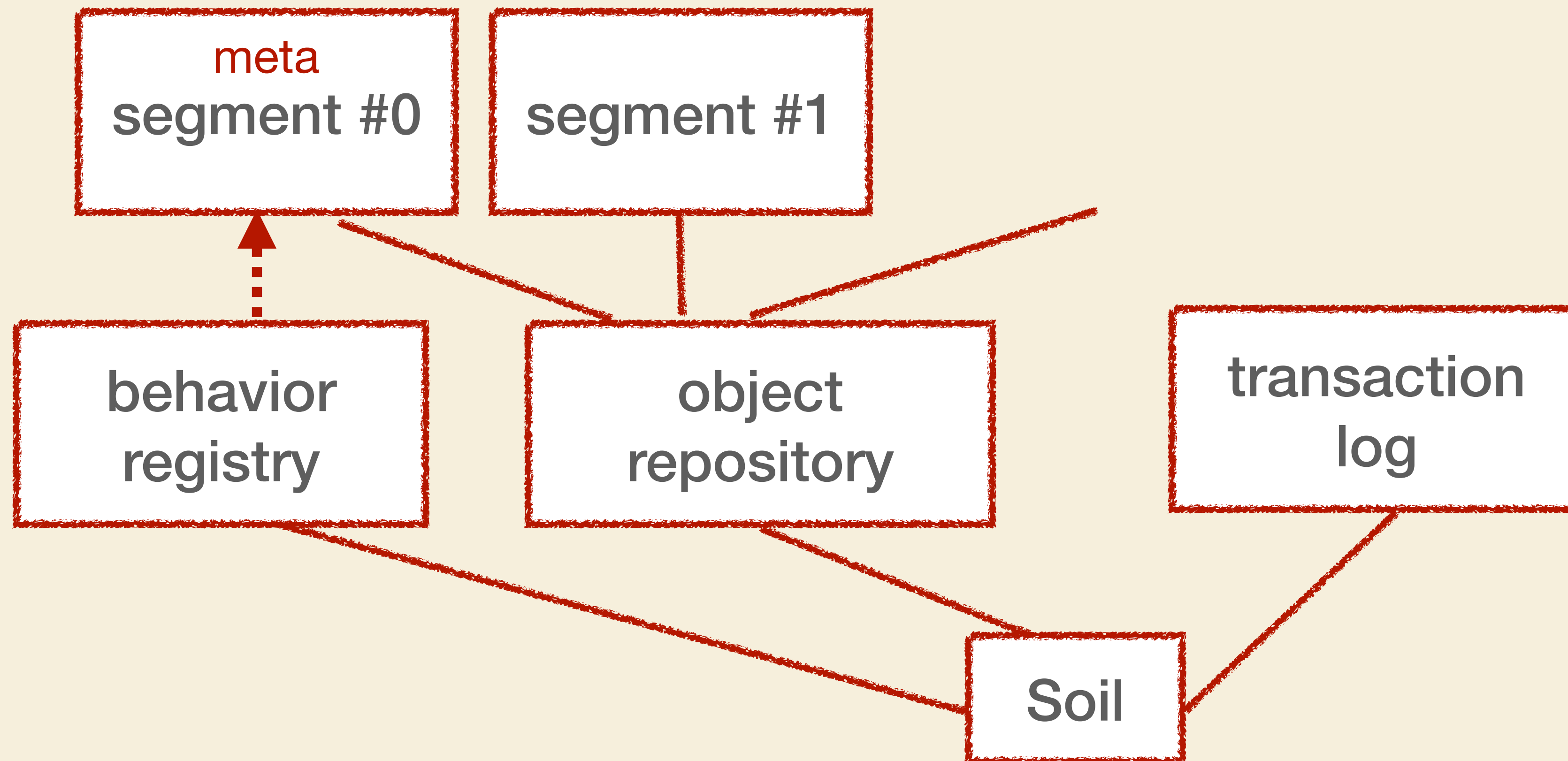
segment files



transaction log



soil structure



thank you! Questions?

Norbert Hartl
norbert@apptivegrid.de

Marcus Denker
marcus.denker@inria.fr