Zope Object Database (ZODB)

Simon Oram (Electrosoup)
simon@electrosoup.co.uk

Prerequisites

- Very basic understanding of Python Classes
- Understanding of dicts
- Understanding of lists

What is ZODB?

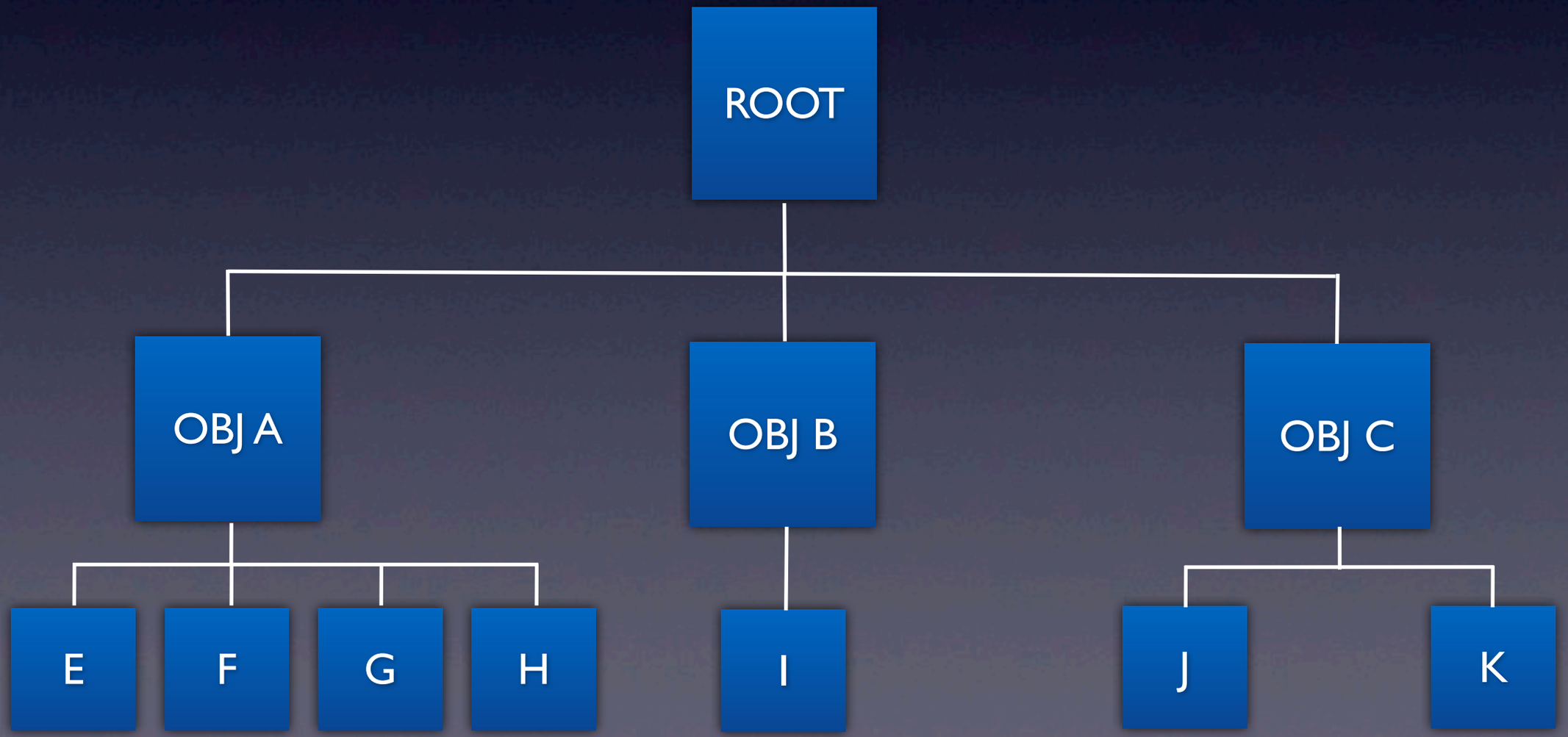
- A technology that persists python objects.
- Over 15 years old, solid battled tested.
- Based on the Pickle module from the standard library
- Can persist any object a pickle can persist

What ZODB offers?

- Caching, transactions, history/undo, Blobs.
- Variety of storage options, File, Memory, Relational Database
- Completely seperate from Zope the Web Framework.
- Grab it from PyPi: pip install ZODB3

Structuring Objects

Data is generally stored hierarchically, similar to your computers file system.



Persisting Objects

- Any ‘pickleable’ object can be persisted
- Must inherit from persistent package
- Has other optimised of lists lists, dicts, sets BTrees etc.

Persisting an Object Example Code

```
>>> from persistent import Persistent
>>> class BlogEntry(Persistent):
>>>     def __init__(self, title, text):
>>>         self.title = title
>>>         self.text = text
```

Connecting to ZODB

```
>>> from ZODB.FileStorage import FileStorage

>>> from ZODB.DB import DB

>>> storage = FileStorage('Data.fs')

>>> db = DB(storage)

>>> connection = db.open()

>>> root = connection.root()

>>> root

{} <-- dict like object
```

Writing to ZODB

```
>>> import transaction

>>> root['A Blog'] = persistent.PersistentList()

>>> an_entry = BlogEntry('Some Title','Some Text')

>>> root['A Blog'].append(an_entry)

>>> transaction.commit()
```

Retreiving an Object

```
>>> from ZODB.FileStorage import FileStorage

>>> from ZODB.DB import DB

>>> storage = FileStorage('Data.fs')

>>> db = DB(storage)

>>> connection = db.open()

>>> root = connection.root()

>>> root[‘A Blog’]

{'A Blog': [<BlogEntry object at ...>]}
```

Modifying an Object

```
>>> blog_entry = root['A Blog'][0]
>>> blog_entry.title = 'New Title'
>>> transaction.commit()
>>> root['A Blog'][0].title
'New Title'
```

Aborting an transaction

```
>>> blog_entry = root['A Blog'][0]
>>> blog_entry.title
'a title'
>>> blog_entry.title = 'Another title'
>>> blog_entry.title
'another title'
>>> transaction.abort()
>>> blog_entry.title
'title'
```

What ZODB doesn't do

- No query language, python IS the query language
- No cataloging and indexing out of the box.
- Only works in CPython
- Only works in Python2.x

Testing

- Superb for unit tests, no need to mock ORM/Database connections.
- Tests are therefore very fast

QUESTIONS?