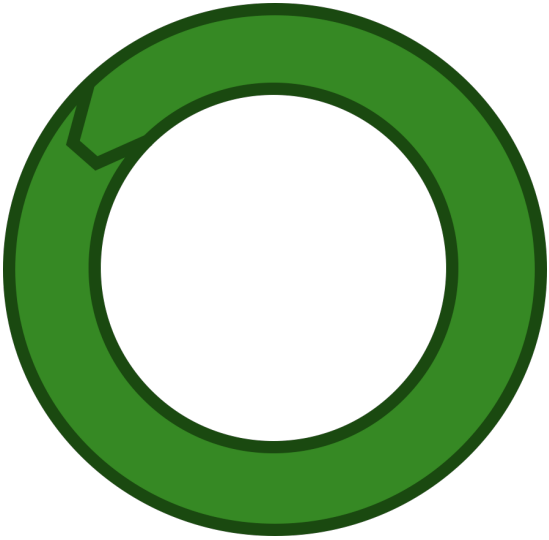
Asko Soukka, asko.soukka@iki.fi
github.com/datakurre

ZODB

—

the Python database

JYVÄSKYLÄN YLIOPISTO



ZODB

—

the Python database

PyCon Finland 31.10.2016

(Zope) Object Database

- Native object tree database for Python since 2002
- ACID transactions with snapshot isolation (MVCC)
- ZODB, persistent, BTrees, transaction, zope.interface, ZConfig, zc.lockfile, zodbpickle (optional: ZEO, zc.zrs)
- Latest version: 5.0.0 (2016-09-06), Python 2.7, 3.4, 3.5

Persisting TaskList with ZODB

Creating new ZODB

In [5]:
```
import ZODB
import ZODB.FileStorage
import transaction

ZODB_FILENAME = 'mydata.fs'

storage = ZODB.FileStorage.FileStorage(ZODB_FILENAME)
db = ZODB.DB(storage)
connection = db.open()
root = connection.root()
```

Analyzing empty ZODB

Crash Course

In [6]:
```
from ZODB.scripts imp
analyze.report(analyze.analyze(ZODB_FILENAME))
```

```
Processed 1 records in 1 transactions
Average record size is   68.00 bytes
Average transaction size is   68.00 bytes
Types used:
Class Name                                        Count     TBytes    Pct AvgSize
------------------------------------------------- ------- --------- ----- -------
persistent.mapping.PersistentMapping                  1         68 100.0%   68.00
================================================= ======= ========= ===== =======
                                Total Transactions        1                  0.07k
                                     Total Records        1        0k 100.0%   68.00
                                   Current Objects        1        0k 100.0%   68.00
```

Persisting TaskList

In [7]:
```
root['tasks'] = TaskList()

import transaction
transaction.commit()
```

ZODB can be a lot of fun

– It's implicit — if object pickles, it persists.

  Just remember to inherit your classes from Persistent.

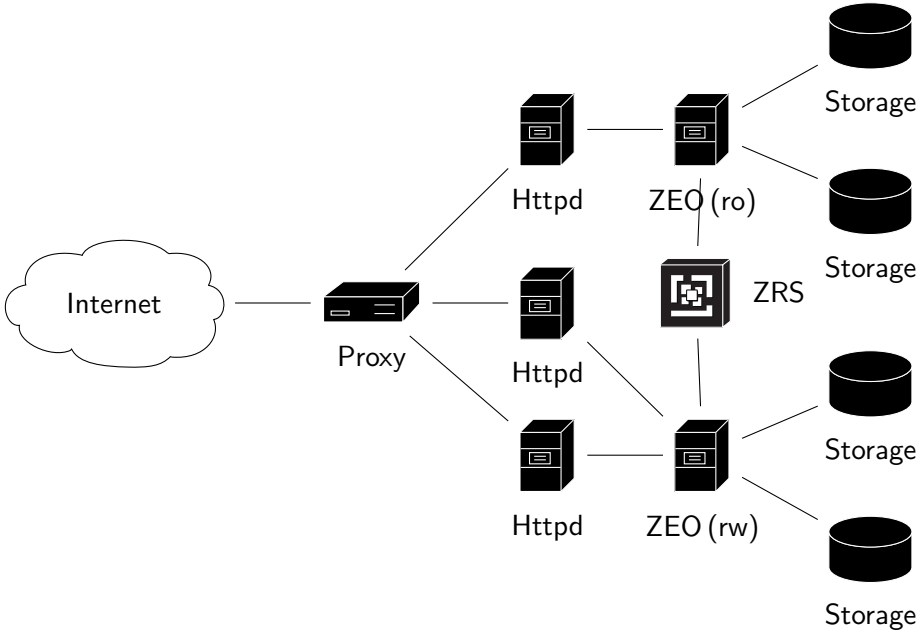
– It's pythonic and does object references right.

– It's safe, append only, immutable, but packs on request.

– It's fast, unless your code slows it down...

Not necessarily *web scale*

- Only one simultaneous request per connection.
  - Each connection has its own *connection cache*.
  - *Connection cache* = unpickled objects in RAM.
  - Must be big enough to store the current working set.
- Search indexes in ZODB are prone to write conflicts.
- No multi-master replication. But see RelStorage and neoppod.


Internet
Proxy
Httpd
Httpd
Httpd
ZEO (ro)
ZRS
ZEO (rw)
Storage
Storage
Storage
Storage

BUILT-IN SECURITY

Make highly granular security assertions against database content, allowing for tight access control of specific resources.

TEXT INDEXING

Built-in text indexing and search allows your users to find their content without requiring you to configure, manage and learn an external text indexing system.

WORKFLOW

Define workflow states and transitions for types of content, and enforce them.

UNDO

Undo virtually any action taken via the management interface or by a retail user.

Substance D

Performance and availability monitoring is baked in. Use any *statsd* provider to set up visibility into your application's performance.

AUDITING

Every action taken by each site user is logged into a built-in audit log facility.

SCREENSHOTS


Questions?
github.com/datakurre/pyconfi2016