

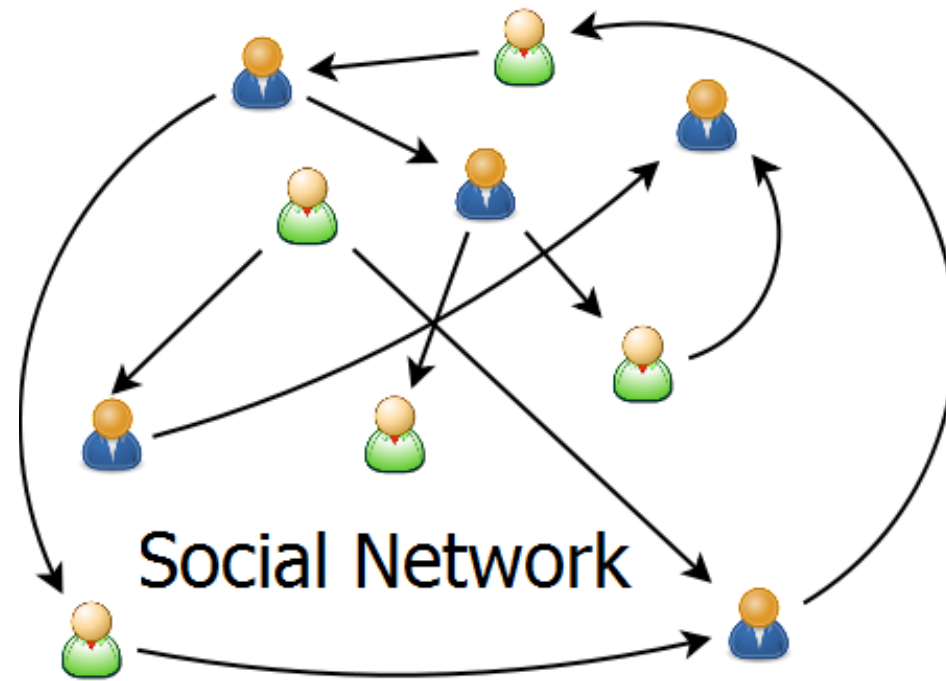


ZODB: A Persistent Graph of Python Objects

By
Christopher Lozinski

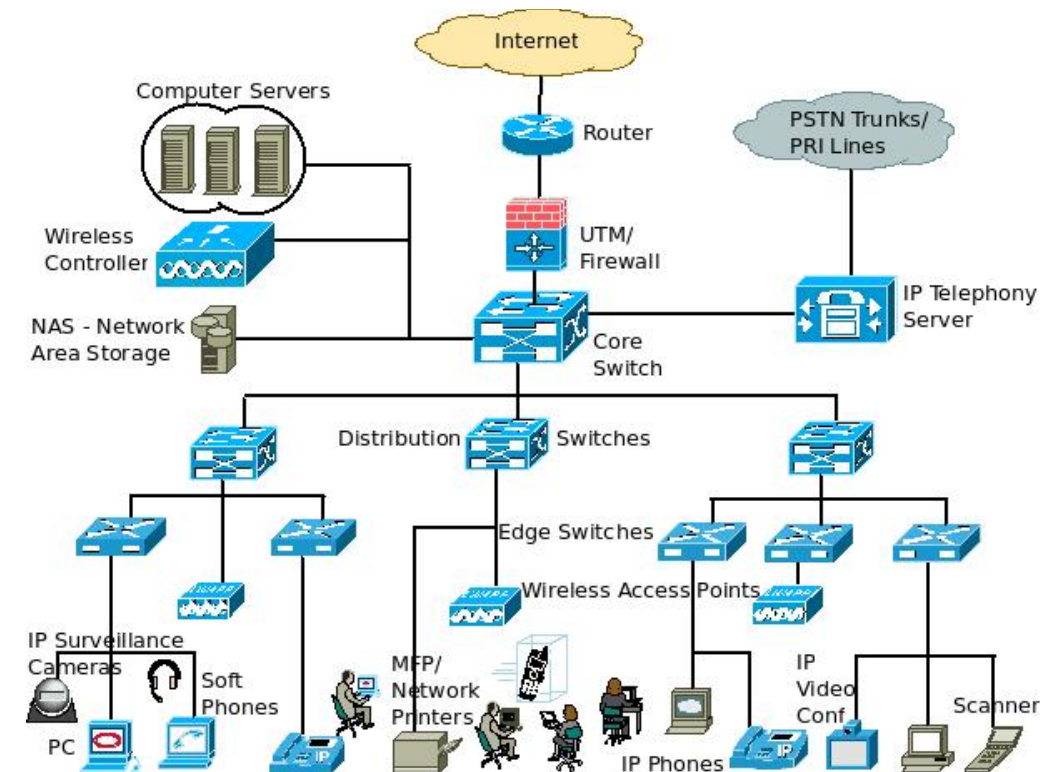
Why Use a Graph Database?

Social Network



© Christopher Lozinski

Computer Network

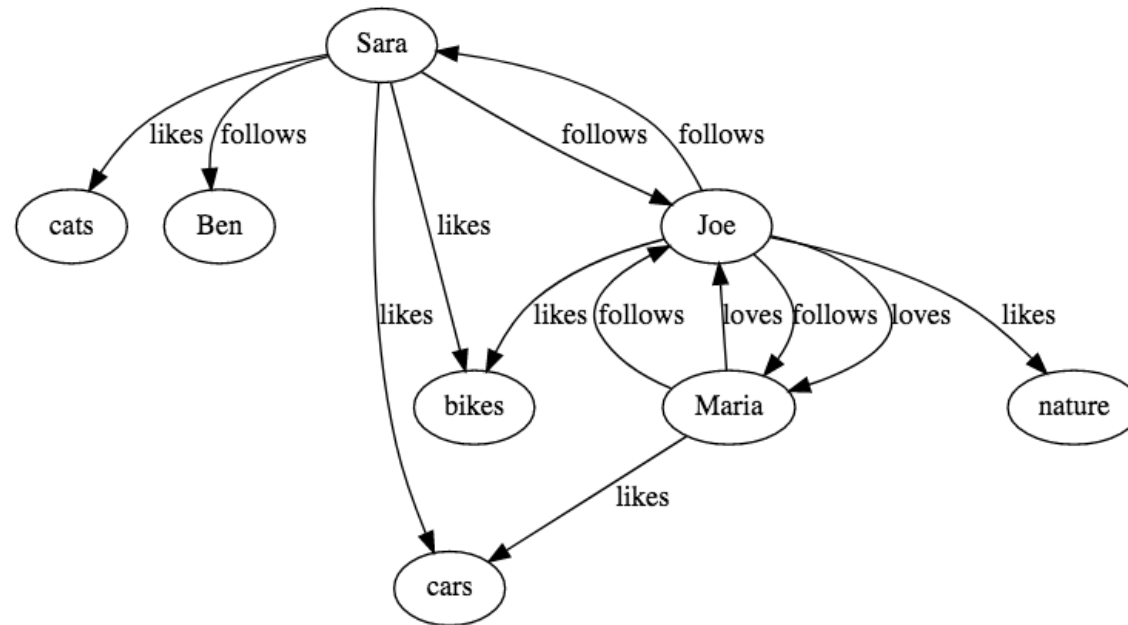


CC BY-NC 3.0 US

t
PythonLinks.info/zodb #2

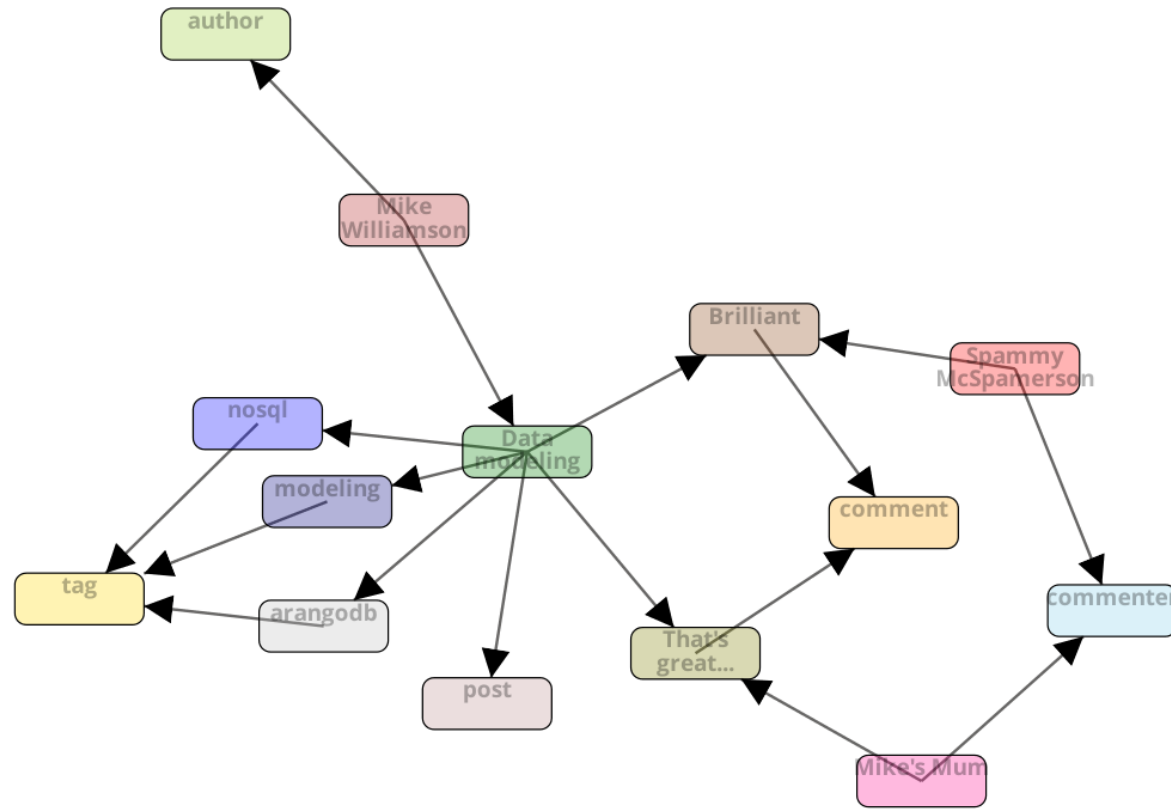
Neo 4J Property Graph Database

Neo, Graphagus

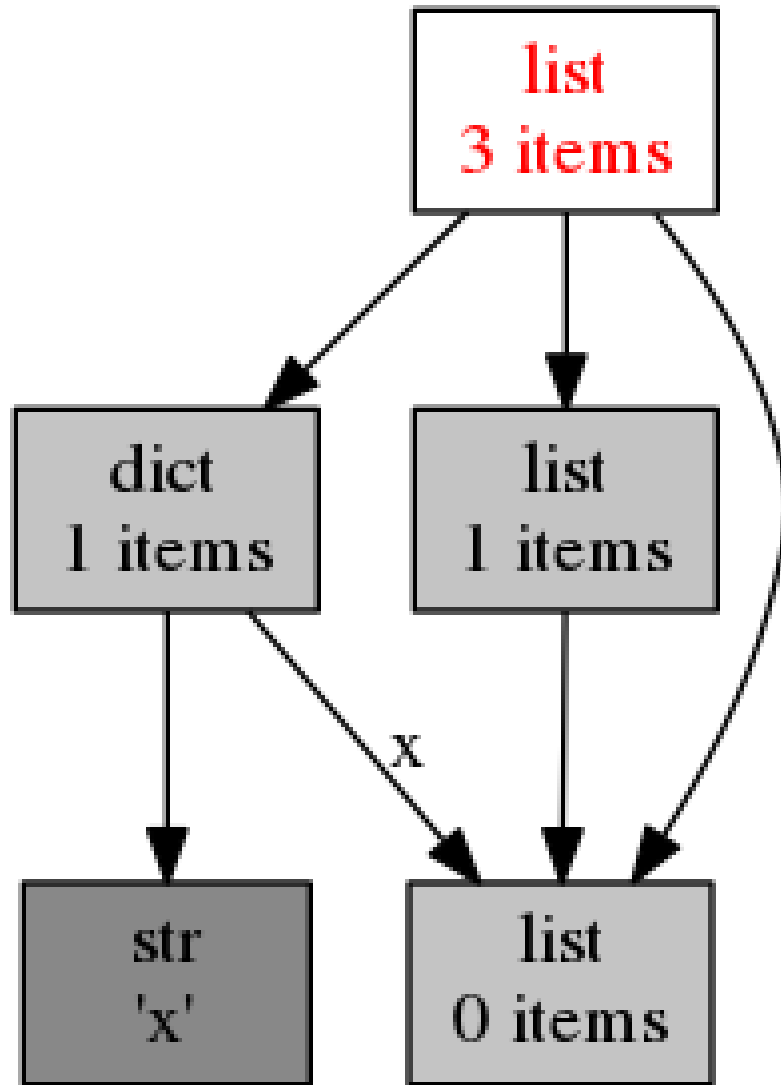


Sara follows Joe.
Sara follows Ben.
Sara likes bikes.
Sara likes cars.
Sara likes cats.
Aria follows Joe.
Maria loves Joe.
Maria likes cars.
Joe follows Sara.
Joe follows Maria.
Joe loves Maria.
Joe likes bikes.
Joe likes nature

ArrangoDB



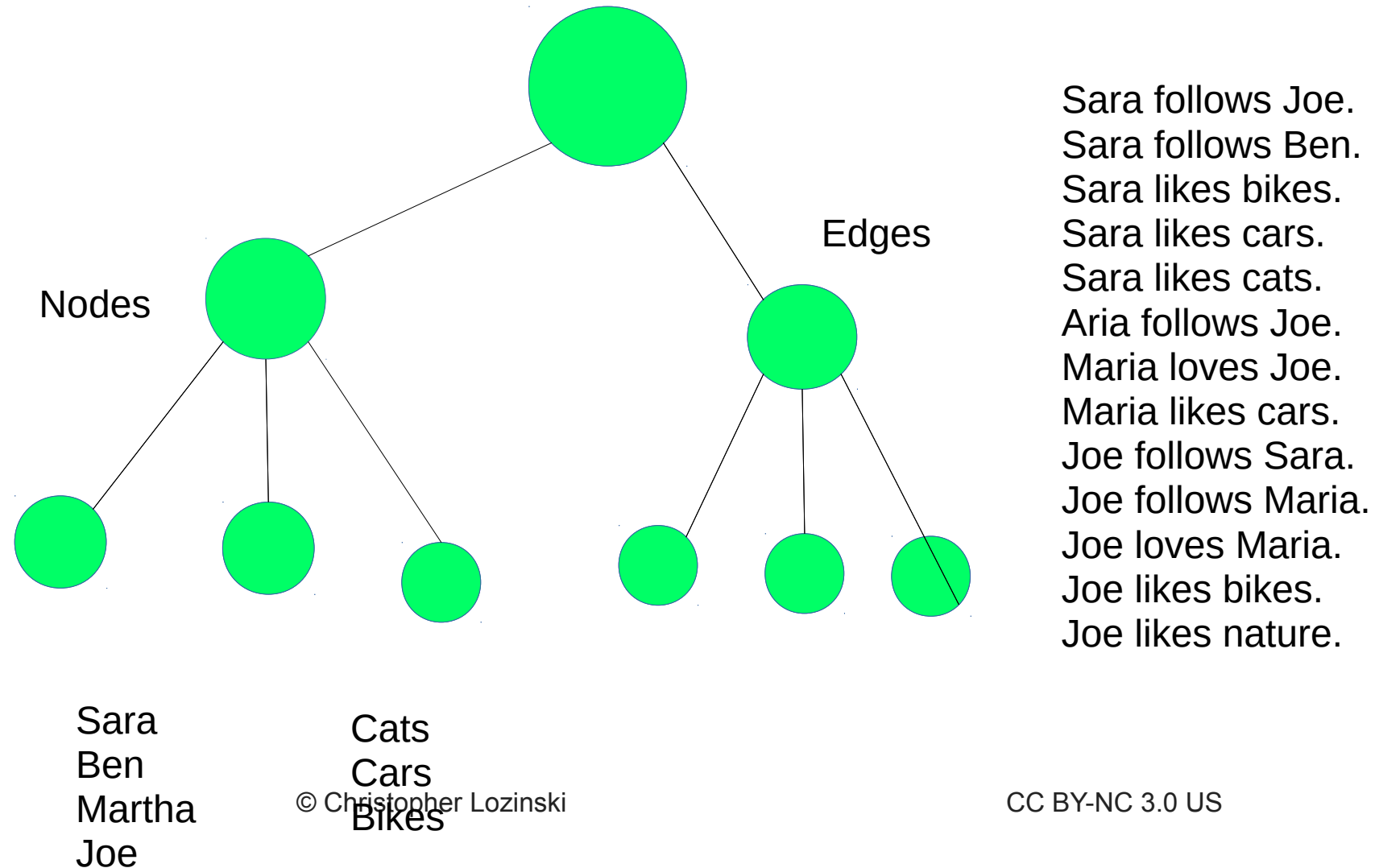
```
{
  title: "Data modelling",
  text: "lorum ipsum...",
  author: "Mike Williamson",
  date: "2015-11-19",
  comments: [
    {
      author: "Mike's Mum",
      email: "mikes_mum@allthemums.com",
      text: "That's great honey",
    },
    {
      "author": "spammer@fakeguccihandbags.com",
      "title": "Brilliant",
      "text": "Gucci handbags...",
    }
  ],
  tags: ["mongodb", "modeling", "nosql"]
}
```



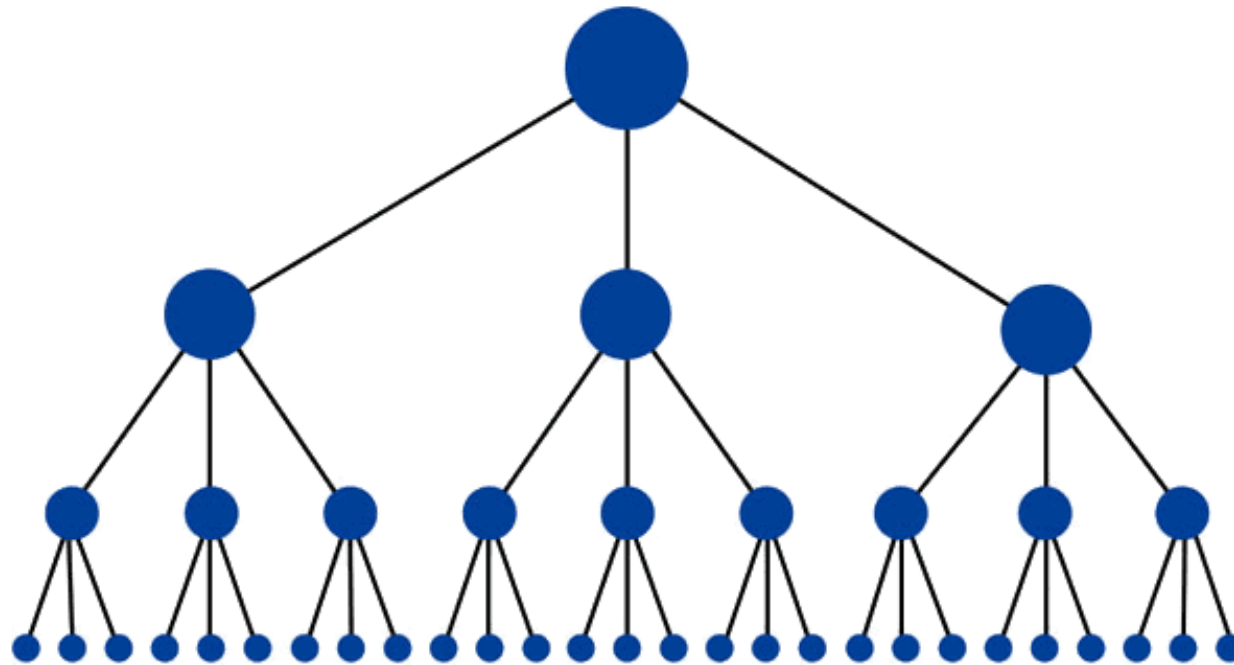
ZODB

Graph of Persistent Python Objects

Graphagus: A Property Graph Database on ZODB



For Managing Complexity



<http://PythonLinks.info/home>

[The best Python Videos organized by Category](#) (1/1) of 251

[The best Videos of PyCon USA 2018 Organized by Category](#) (1/3) of 174

The best Videos of PyCon USA 2018 Organized by Category

Category: home

Here you have the best rated videos from the conference. Swipe or arrow down for the best talks in each category.

[Search](#) [Profile](#) [Sign Out](#) or swipe ↑ ↓ ←

Score

10 Best Talks in this Category

100%	108	0	Big-O: How Code Slows as Data Grows
100%	58	0	Easy 2D Game Creation With Arcade
100%	58	0	How Netflix does failovers in 7 minutes flat
100%	46	0	Keynote
100%	44	0	Taking Django Async
100%	42	0	Get your resources faster, with importlib.resources
100%	39	0	How to Write Deployment-friendly Applications
100%	36	0	A Bit about Bytes: Understanding Python Bytecode
100%	36	0	Analyzing Data: What pandas and SQL Taught Me About Taking an Average
100%	36	0	Inside the Cheeseshop: How Python Packaging Works

Child Categories

[Talks about Data Science](#) (28)

[Talks about artificial intelligence](#) (6)

[Distributed Systems](#) (15)

[Keynote, Concluding Remarks, and Lightning Talks](#) (8)

[Python in the Organization, and in the communittee](#) (13)

[Python Skills Development](#) (40)

[Python Software](#) (58)

[Talks related to Testing](#) (5)

(Reported from the next page)

How to Use the ZODB

So Easy to use!

```
import persistent

class TreeLeaf(persistent.Persistent):

    def __init__(self, title=''):
        self.title=title

    def render(self):
        return self.title
```

ZODB Tutorial: Add Leaf Objects

Single Leaf Object	Multiple LEAF Objects
<pre>#CREATE A SINGLE LEAF OBJECT leaf = TreeLeaf('Leaf') root.leaf=leaf</pre>	<pre>#CREATE MULTIPLE LEAF OBJECTS Leaf1 = TreeLeaf('Green Leaf') leaf2 = TreeLeaf('Red Leaf') #ADD THEM TO THE ROOT root['leaf1'] = leaf1 root['leaf2'] = leaf2</pre>

ZODB Tutorial: Create a Database

<http://www.zodb.org/en/latest/tutorial.html>

```
import ZODB, ZODB.FileStorage
```

```
db = ZODB.db('Data.fs')
```

```
connection = db.open()
```

```
root = connection.root
```

```
#DO SOMETHING
```

```
transaction.commit()
```

It is just Python

(No SQL SELECT)

```
#UPDATE THE Leaf
```

```
root['leaf1'].title="Yellow Leaf"
```

```
transaction.commit()
```

```
#STUPID QUERY
```

```
For key, item in root.items():
```

```
    print (key, item)
```

```
# DELETE AN OBJECT
```

```
del root['leaf1']
```

```
transaction.commit()
```

ZODB is Magical

Creates the illusion that your Python Objects are Persistent



© Christopher Lozinski

CC BY-NC 3.0 US

PythonLinks.info/zodb #13

ZODB is a graph databases

```
#CREATE THE OBJECTS
Leaf1 = TreeLeaf('Green Leaf')
leaf2 = TreeLeaf('Red Leaf')

#ADD THEM TO THE ROOT
root['leaf1'] = leaf1
root['leaf2'] = leaf2
```

```
#IT IS A GRAPH DATABASE
leaf1.sibling = leaf2
leaf2.sibling = leaf1
transaction.commit()
```

Hierarchical Calculations

```
class Video(Persistent):  
    def countLeaves(self):  
        return 1
```

```
class Category(Container):  
    def countLeaves(self):  
        total=0  
        for item in self.values():  
            total+=item.countLeaves()  
        self.branchSize = total  
        return total
```

The Best Videos on any branch of the tree

}}

Score	Votes	10 Best Python Videos Overall
100%	93👍	Interactive 3D Visualization in Jupyter
100%	77👍	Practical Sphinx
100%	76👍	Taking Django Async
100%	73👍	Hands-on introduction to Deep Learning with Keras and Tensorflow
100%	63👍	Conversational AI with Rasa Core & NLU
100%	61👍	PyViz: Dashboards for Visualizing 1 Billion Datapoints in 30 Lines of Python
100%	52👍	Inside the Cheeseshop: How Python Packaging Works (PyCon US 2018)
100%	52👍	Optimization with Cython: Ising Models (Part 2)
100%	47👍	Building new NLP solutions with spaCy and Prodigy (PyData Berlin)

ZODB Uses Pickle

Pickle

Module for (de)serialization: Storing complete Python objects into files and later loading them back.

- Supports almost all data types – good.
- Works only with Python – bad.



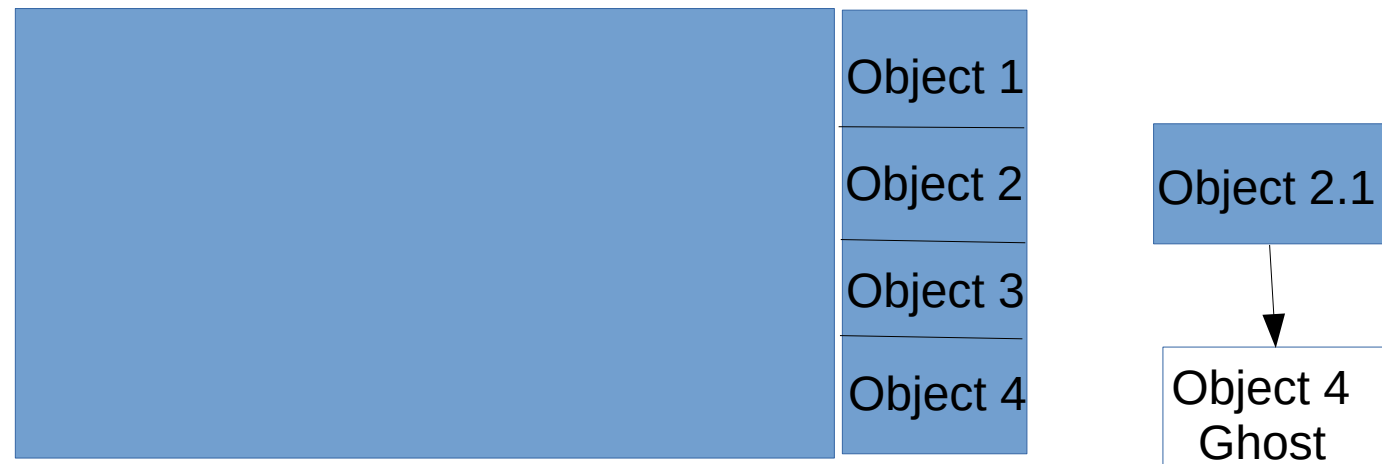
```
import pickle
pickle.dump(object, openBinaryFile) # Save object to an open file
object = pickle.load(openBinaryFile) # Restore an object from an open file
```

File Storage

Objects are Written to the end of a File



Ghost Objects



FileStorage Transactions

Object 1	Object 5	Object A	Object D
Object 2	Object 3	Object B	Object E
Object 3	Object 6	Object 3	Object F
Object 4	Object 7	Object C	Object G

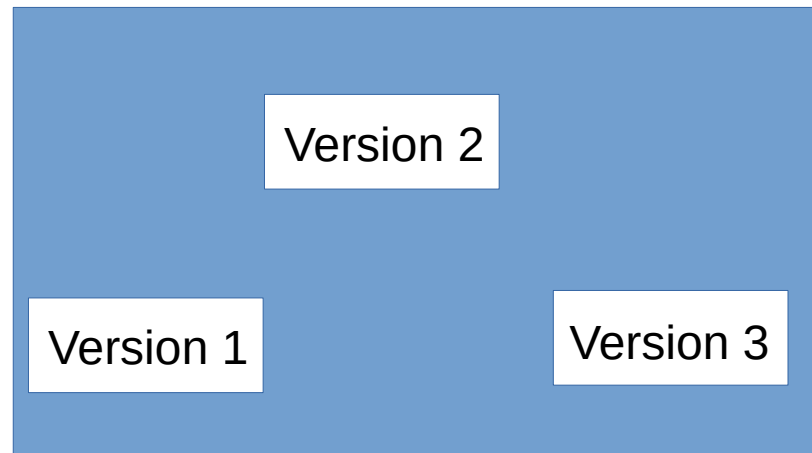
Transaction 1

Transaction 3

Transaction 2

Transaction 4

Object Versions

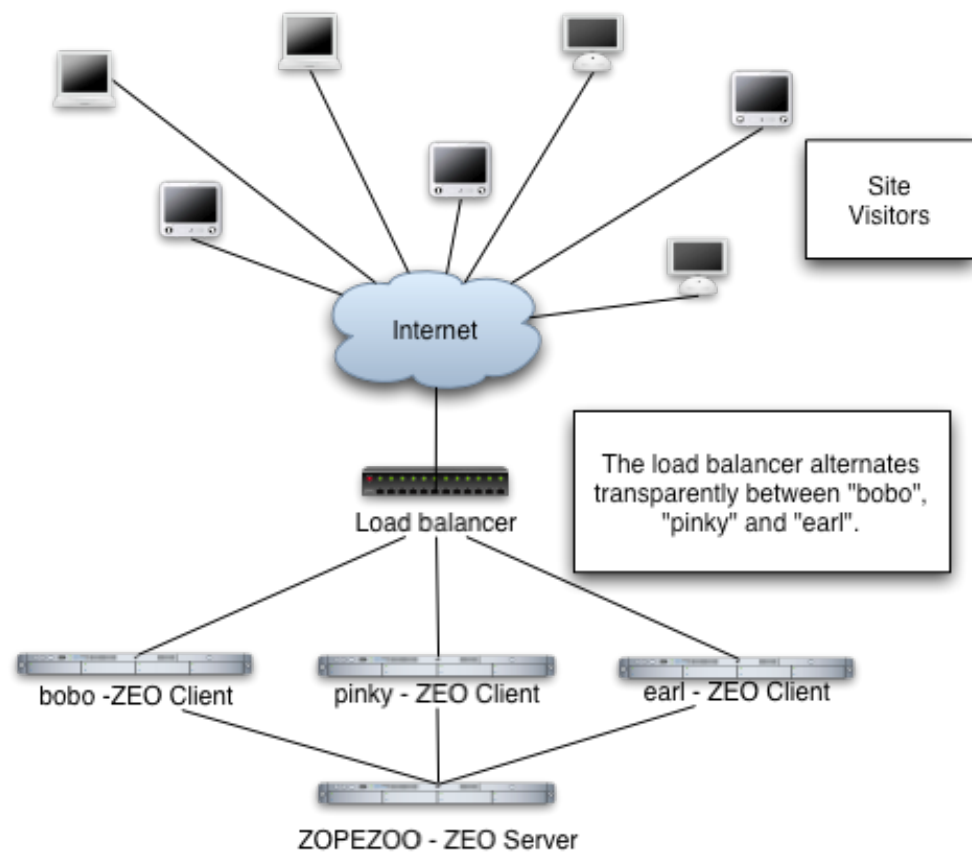


Rel Storage

Storing ZODB Objects in a Relational Database.

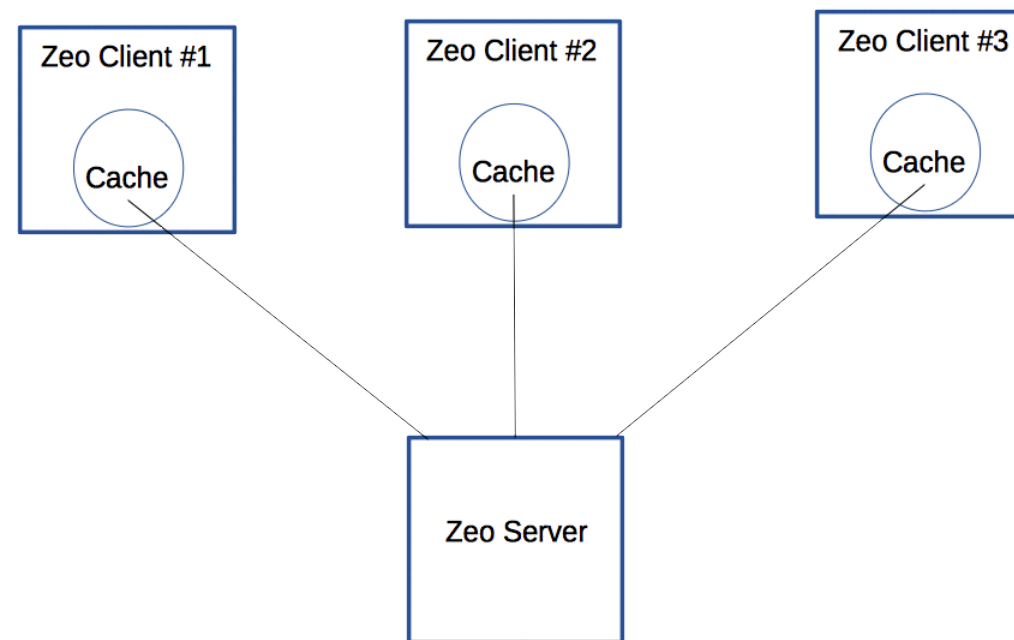
PostgreSQL, Oracle, MySQL

Object Id	Version Number	Pickle
-----------	----------------	--------



ZEO

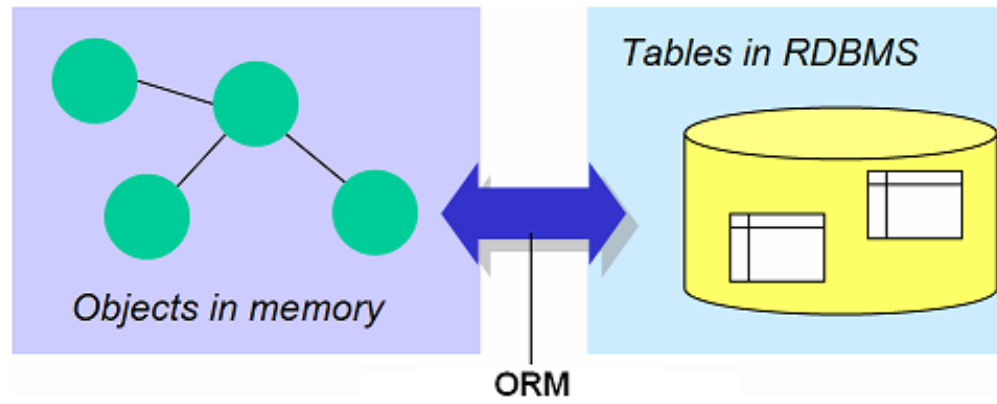
Client Server Storage



ZODB Advantages

by Jim Fulton

-
-



No Database to administer

No Database Administrator

No Database Schema

No ORM

No Referential Integrity Problems

Automatic Garbage Collection

No manual reads and writes



Flask ZODB



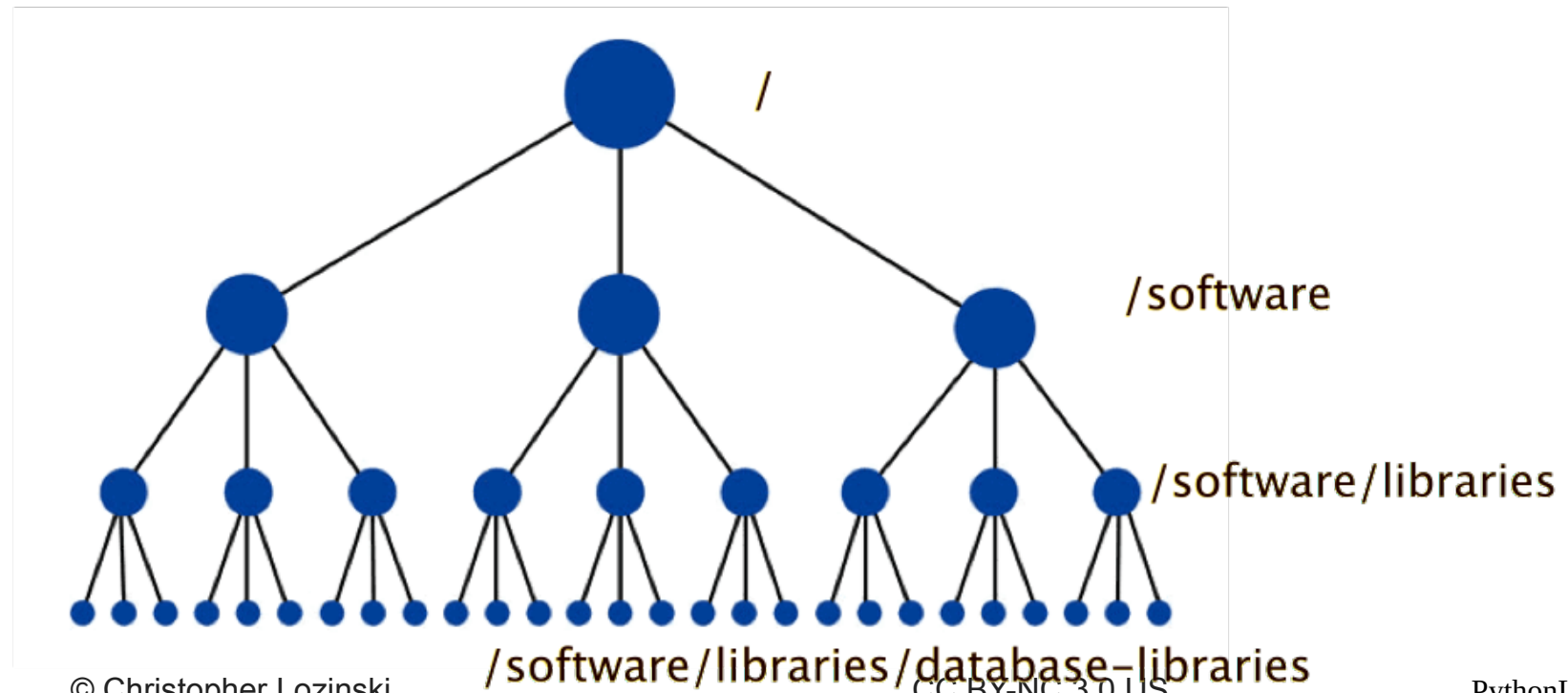
Pyramid™



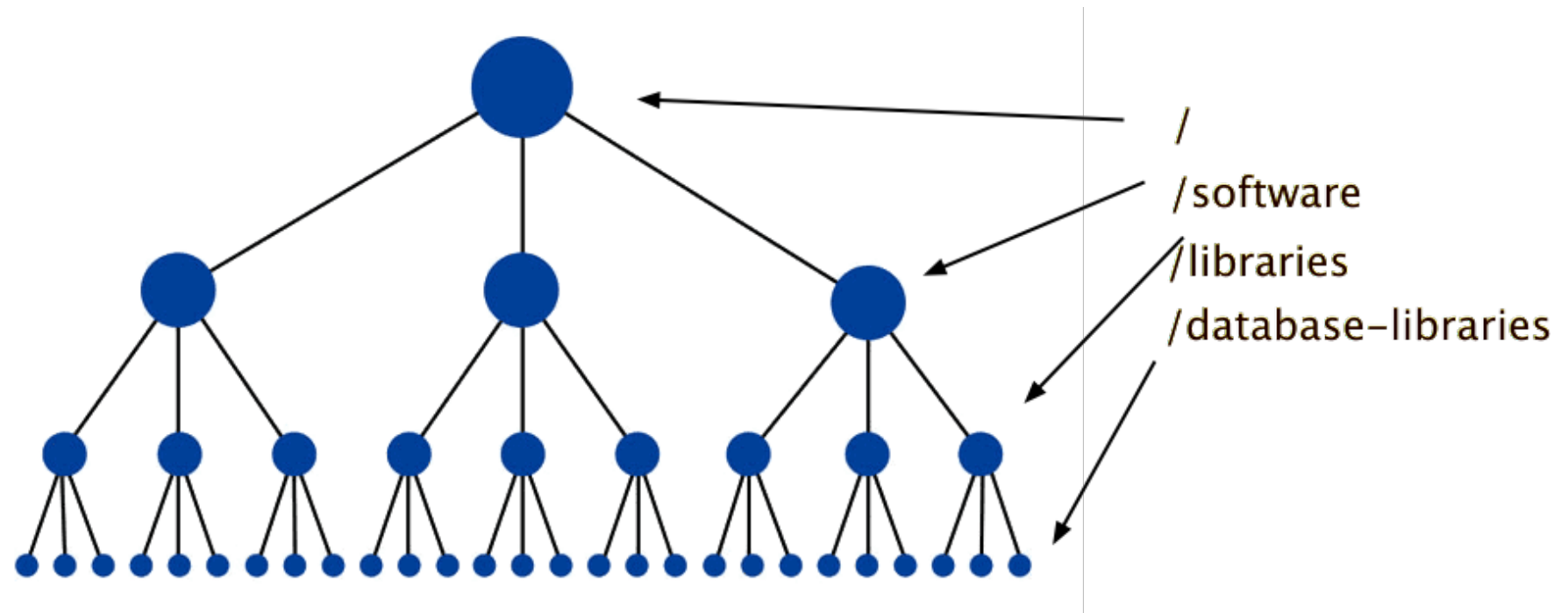
Dolmen Cromlech

Traversal

/ Software / Libraries / Database Libraries



Zopache Canonical URL's



ZODB Cell Phone Interface

<http://PythonLinks.info/>

The best Python Videos organized by Category (1/1) 1 🏆

The best Videos of PyCon USA 2018 Organized by Category (1/2) 1 🏆

The best Videos of PyCon USA 2018 Organized by Category

[Manage CkEdit](#)

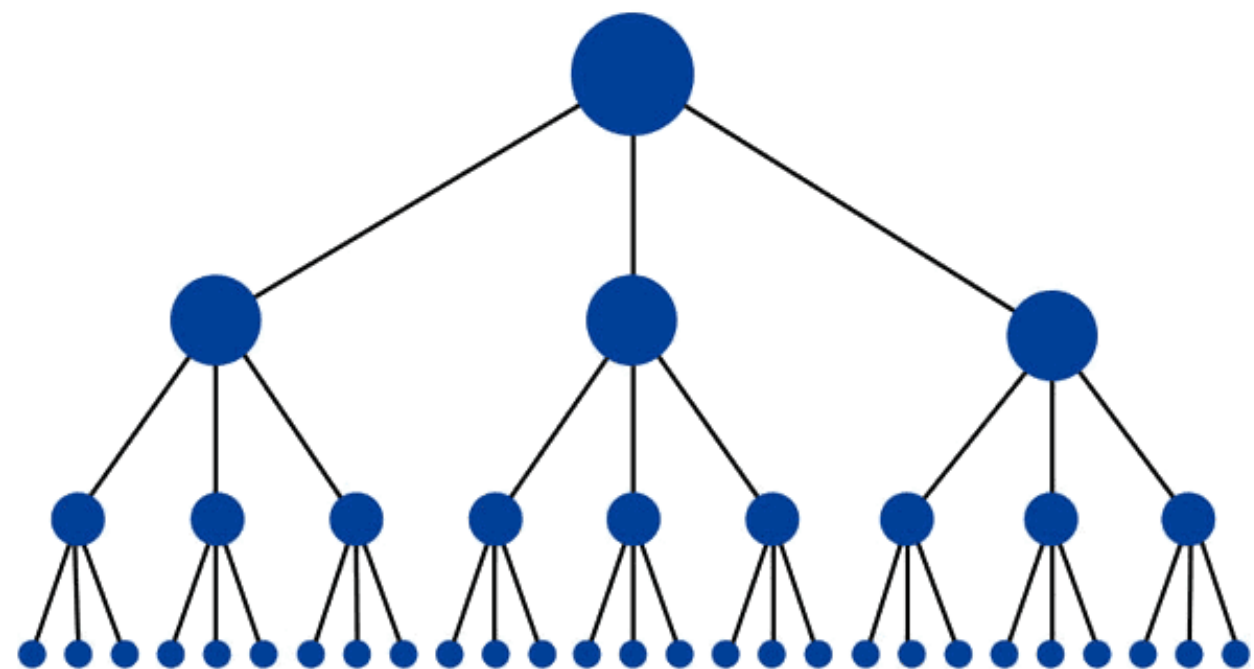
Category: home

Here you have the best rated videos from the conference. Swipe or arrow down for the best talks in each category.

or swipe ↑ ↓ ←

Score	10 Best Talks in this Category	
100% 66 🍑	0 🍑	Big-O: How Code Slows as Data Grows
100% 56 🍑	0 🍑	Performance Python: Seven Strategies for Optimizing Your Numerical Code
100% 46 🍑	0 🍑	Easy 2D Game Creation With Arcade
100% 35 🍑	0 🍑	Taking Django Async
100% 31 🍑	0 🍑	Get your resources faster, with importlib.resources
100% 29 🍑	0 🍑	How to Write Deployment-friendly Applications
100% 28 🍑	0 🍑	Keynote
100% 26 🍑	0 🍑	How Netflix does failovers in 7 minutes flat
100% 25 🍑	0 🍑	A Bit about Bytes: Understanding Python Bytecode

ZODB vs Relational Databases



Students Table

Student	ID *
John Smith	084
Jane Bloggs	100
John Smith	182
Mark Antony	219

Activities Table

ID*	Activity1	Cost1	Activity2	Cost2
084	Tennis	\$36	Swimming	\$17
100	Squash	\$40	Swimming	\$17
182	Tennis	\$36		
219	Swimming	\$15	Golf	\$47

ZODB vs Relational Databases

ZODB	Relational Databases
for item in node: print (item)	You have to do a database join across every single table. 10 Tables: App, Category, City, Company, Country, iFrame Link, Job, Link, Region, Product, and Video.
Zcatalog, Hypatia, repoze.catalog	Select statement

CREATING INDEXES

Using Repoze.catalog

```
1) catalog = Catalog()

2) def get_area(object, default):
    return getattr(object, 'area', default)

3) catalog['area'] = CatalogFieldIndex(get_area)

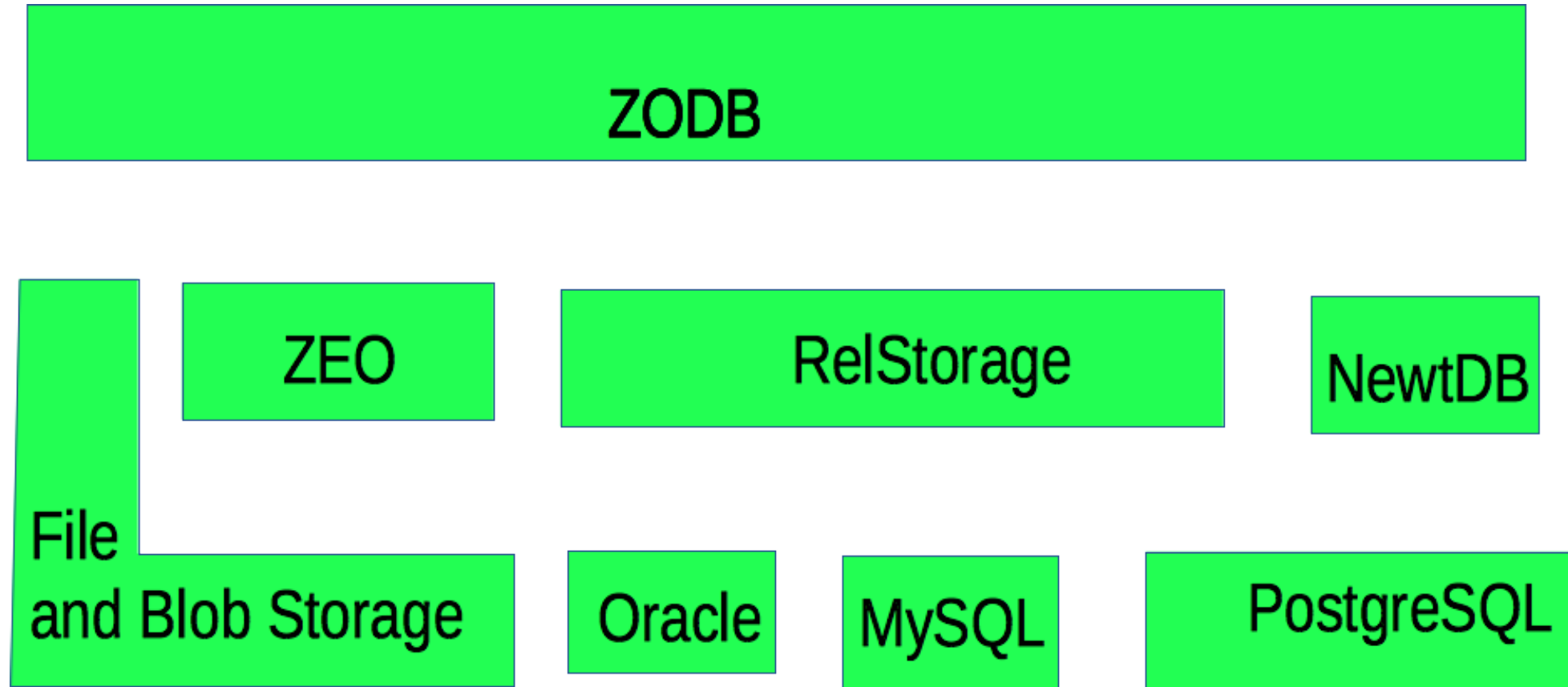
4) leaf-1 = Leaf(area=20)

5) catalog.index_doc(1, leaf-1)

6) catalog.reIndex (1,leaf-1)

7) numdocs, results = catalog.query(Range(20,40))
```

ZODB Storage Options



NewtDB

Persistent Python + PostgreSQL Indexes



Plone ZODB Users

US Plone Sites

- Federal Bureau of Investigation (FBI)
- Central Intelligence Agency (CIA)
- Intellectual Property Rights Center
- US Department of Energy
- USDA Forest Service
- Fermi National Accelerator Lab (Fermilab)
- NASA Science
- Continental Airlines
- UCLA
- Yale University
- Harvard
- The Pennsylvania State University
- University of Notre Dame
- University of Virginia
- University of California - Davis
- University of North Carolina
- University of Louisville
- Novell
- Akamai
- eBay
- Google
- Walmart
- Marriott
- ...and many more.

Worldwide Plone Sites

- Brazilian Government
- 2016 Olympics Brazil
- The British Postal Museum and Archive
- The New Zealand Treasury
- Konica Minolta Printers - Australia
- National Sports Council - Spain
- National Library of South Africa
- University of Oxford
- University of Toronto
- Academy of Performing Arts - Prague
- Open Society Foundation
- Amnesty International
- OXFAM
- Lufthansa
- Nokia
- Clean Clothes Campaign
- RIPE
- Cambridge University
- Royal College of Surgeons
- Oxford University Clinical Academic Graduate School
- ... and many more



Speed

by Jim Fulton

1000's of Transactions per second

For simple transactions relational databases are slightly faster

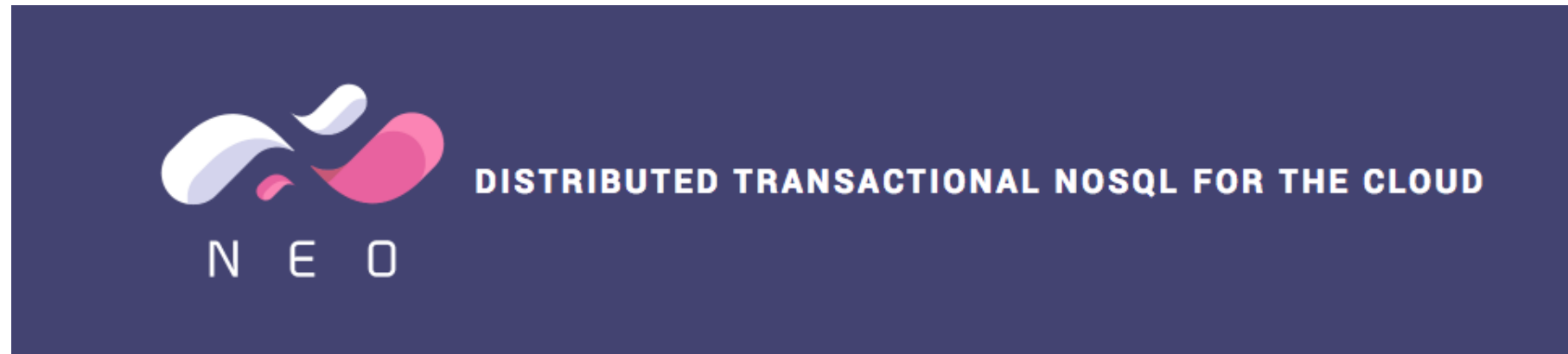
For complex transactions ZODB is faster.

Try to write 10 different classes, in ZODB just append to a file, an RDB requires 10 different writes.

Scalability

Several hundred newspaper content-management systems and web sites were hosted using a multi-database configuration with most data in a main database and a catalog database. The databases had several hundred gigabytes of ordinary database records plus multiple terabytes of blob data.

For larger systems move to NEO (neo.nexedi.com). Up to 80TB in production. 150TB (and growing) in test. Of course it takes time to move that much data around.



Number of Objects

18,446,744,073,709,551,616 Objects

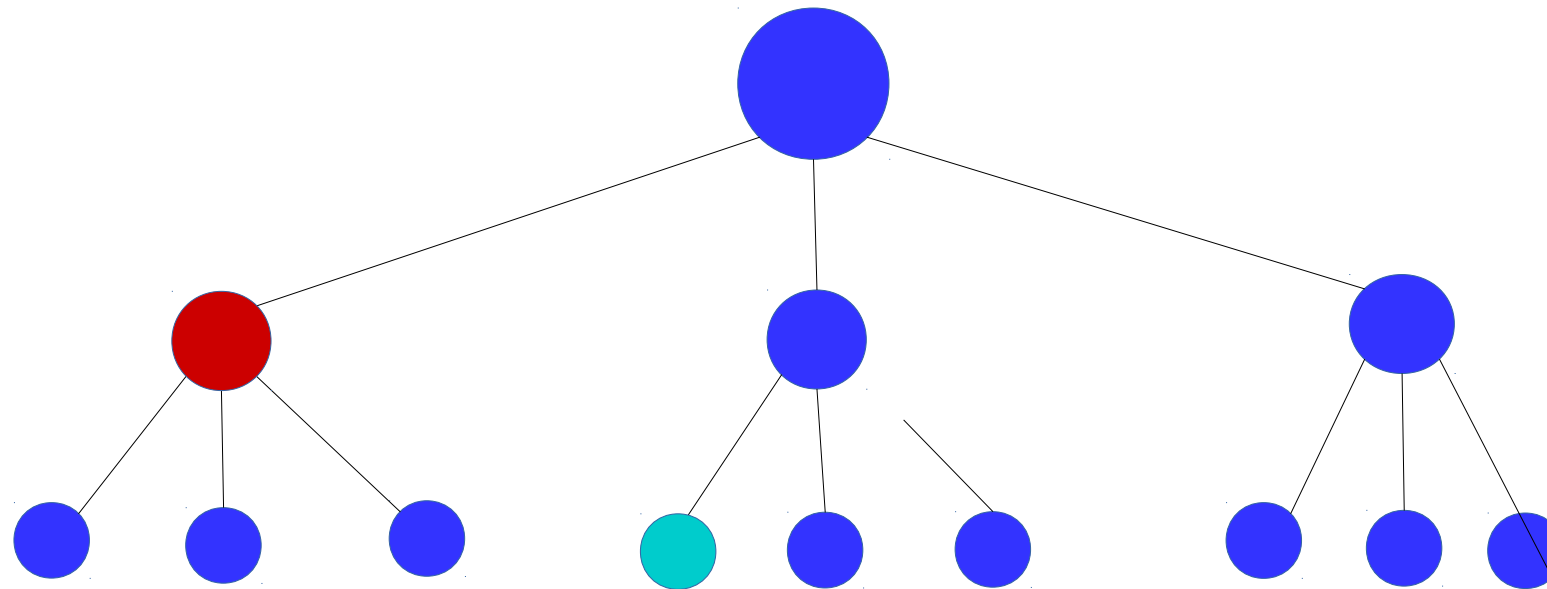
Security

ZeroDB Decrypts Everything on the Client.



MQTT Pub/Sub For Real-Time Chat

MQTT.org



Subscribers



ZODB is for Small Companies

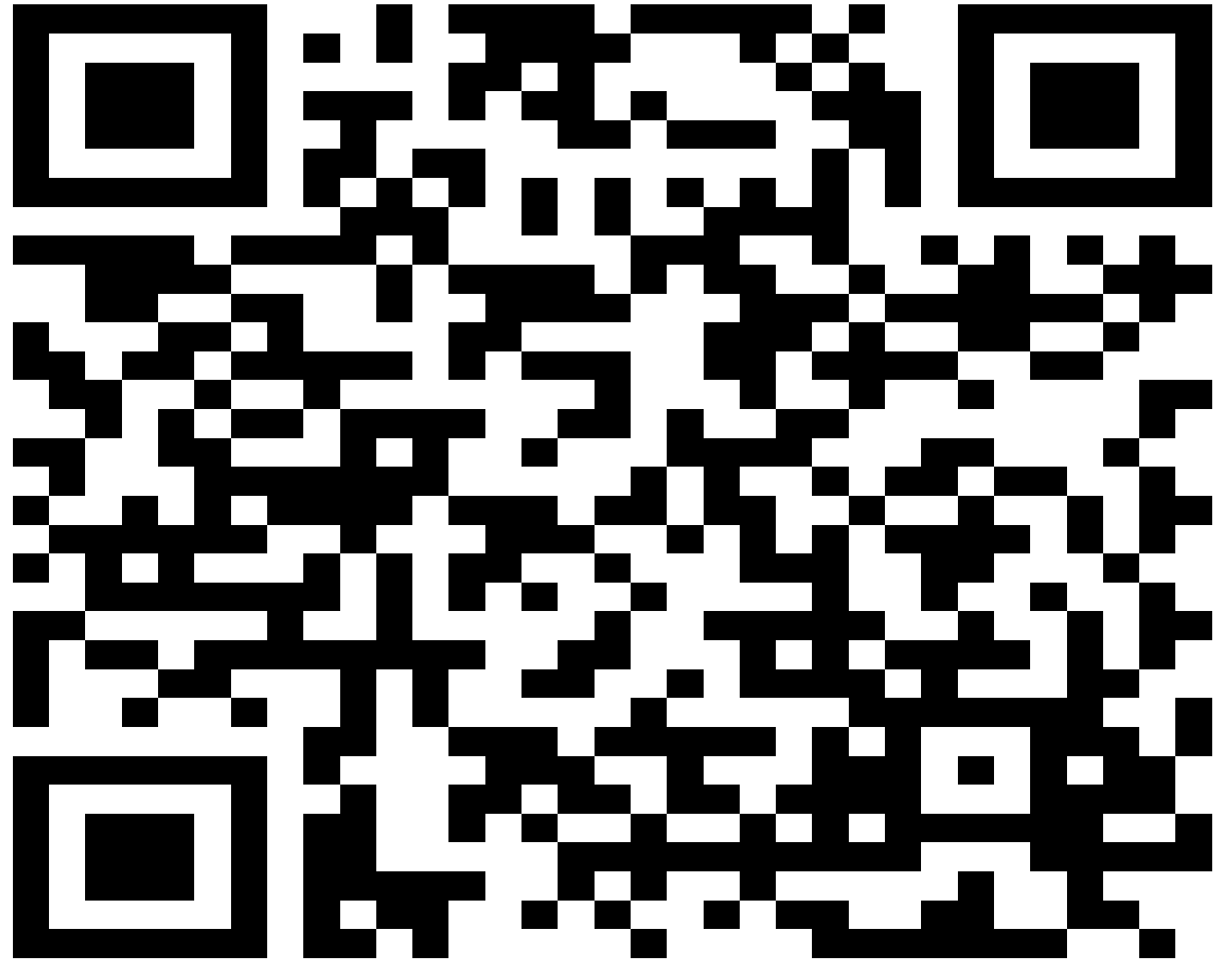
Big Companies	Small Companies
Big Data	Quality Data
Database-Centric	Object-Centric
Manage Complexity	KISS. Pug
Armies of Transient Developers	One Senior Dedicated Developer
Fad Driven	Technology Driven

Questions ?

Follow @PythonLinks
on Twitter



© Christopher Loz



Contact Information

Follow @PythonLinks on Twitter

Christopher Lozinski

[Http://PythonLinks.info](http://PythonLinks.info)

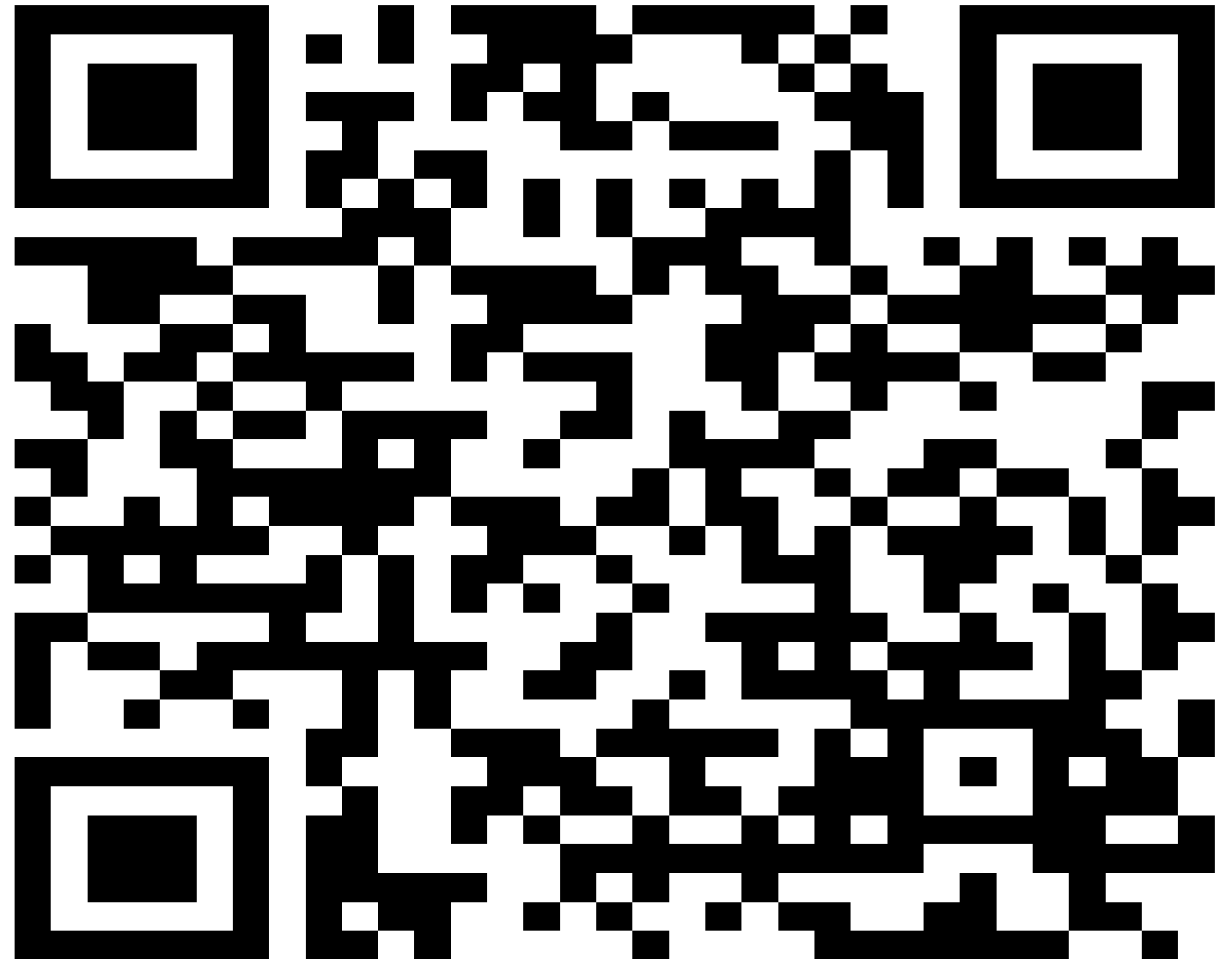
E-Mail: lozinski@PythonLinks.info

Twitter: @PythonLinks

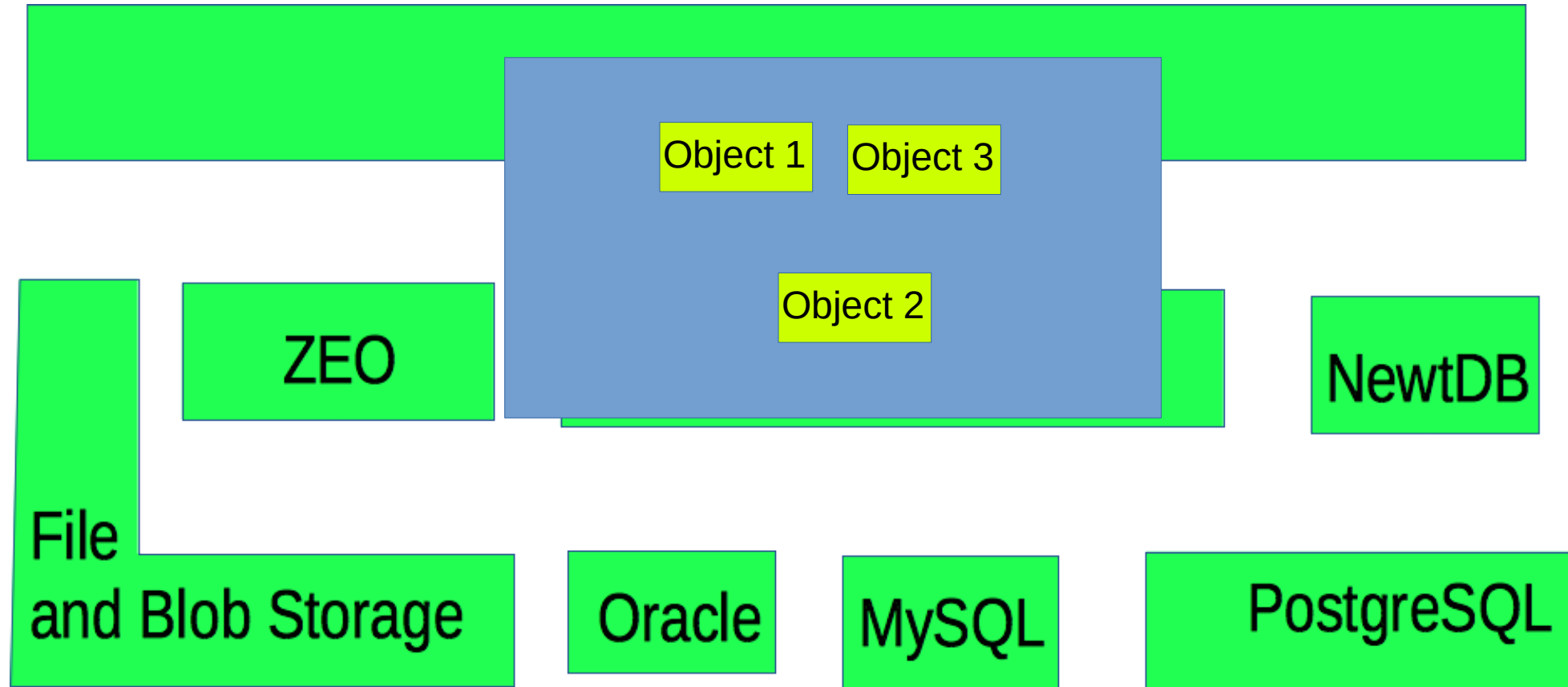
Skype: clozinski

US Phone: +1 (650) 614 1836

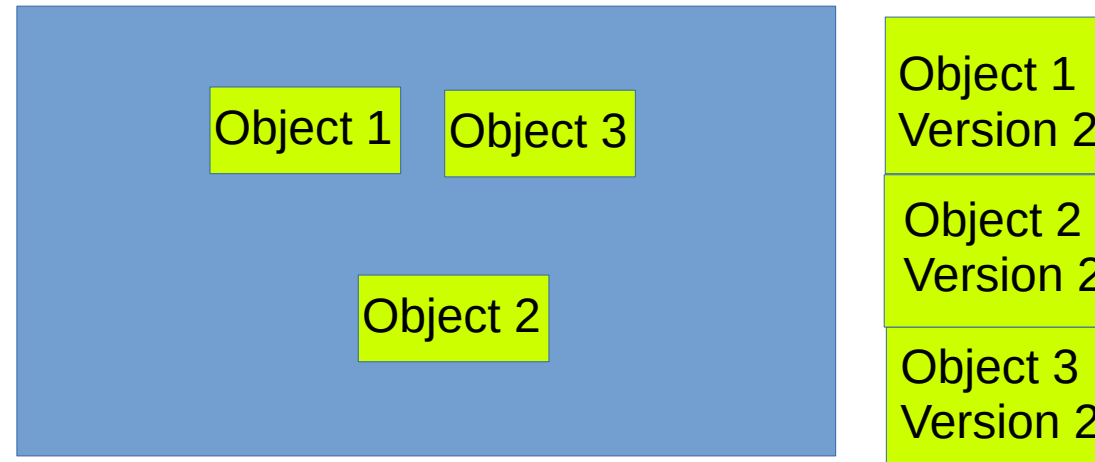
EU Phone: +48 12 361 3136



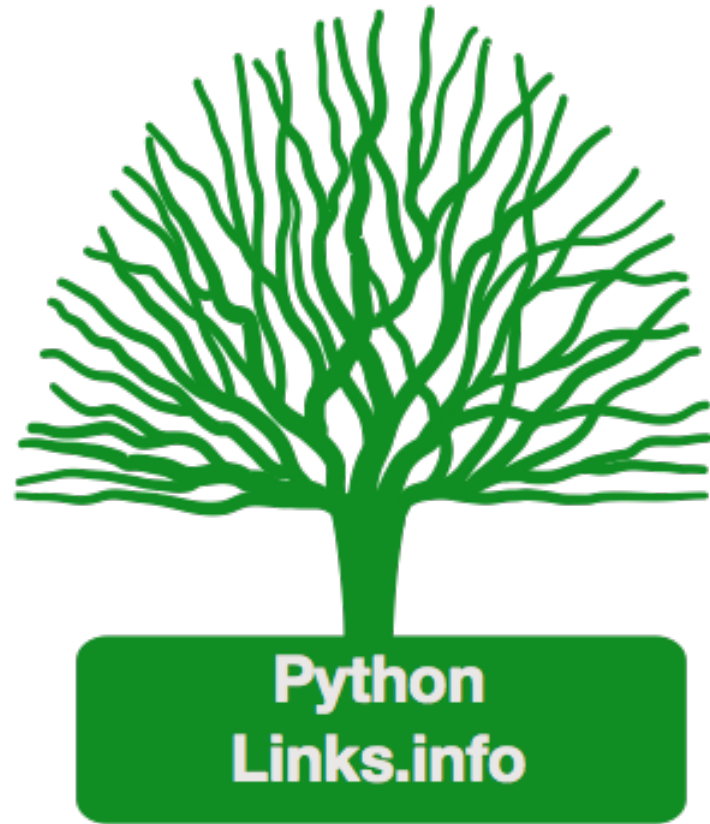
Storing Chat Logs



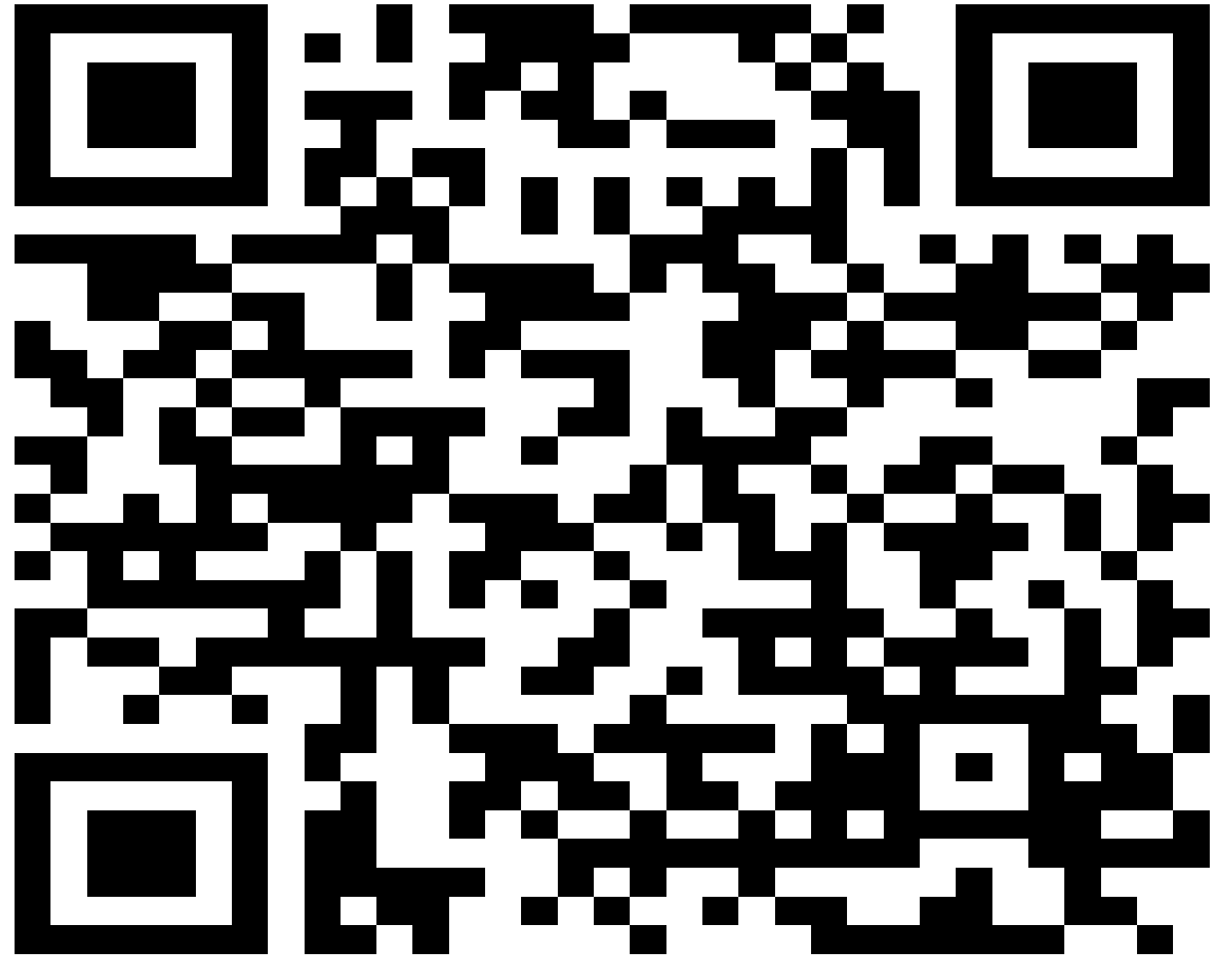
Chat Logs After Batching



Follow @PythonLinks
on Twitter



© Christopher Lozi



Contact Information

Follow @PythonLinks on Twitter

Christopher Lozinski

[Http://PythonLinks.info](http://PythonLinks.info)

E-Mail: lozinski@PythonLinks.info

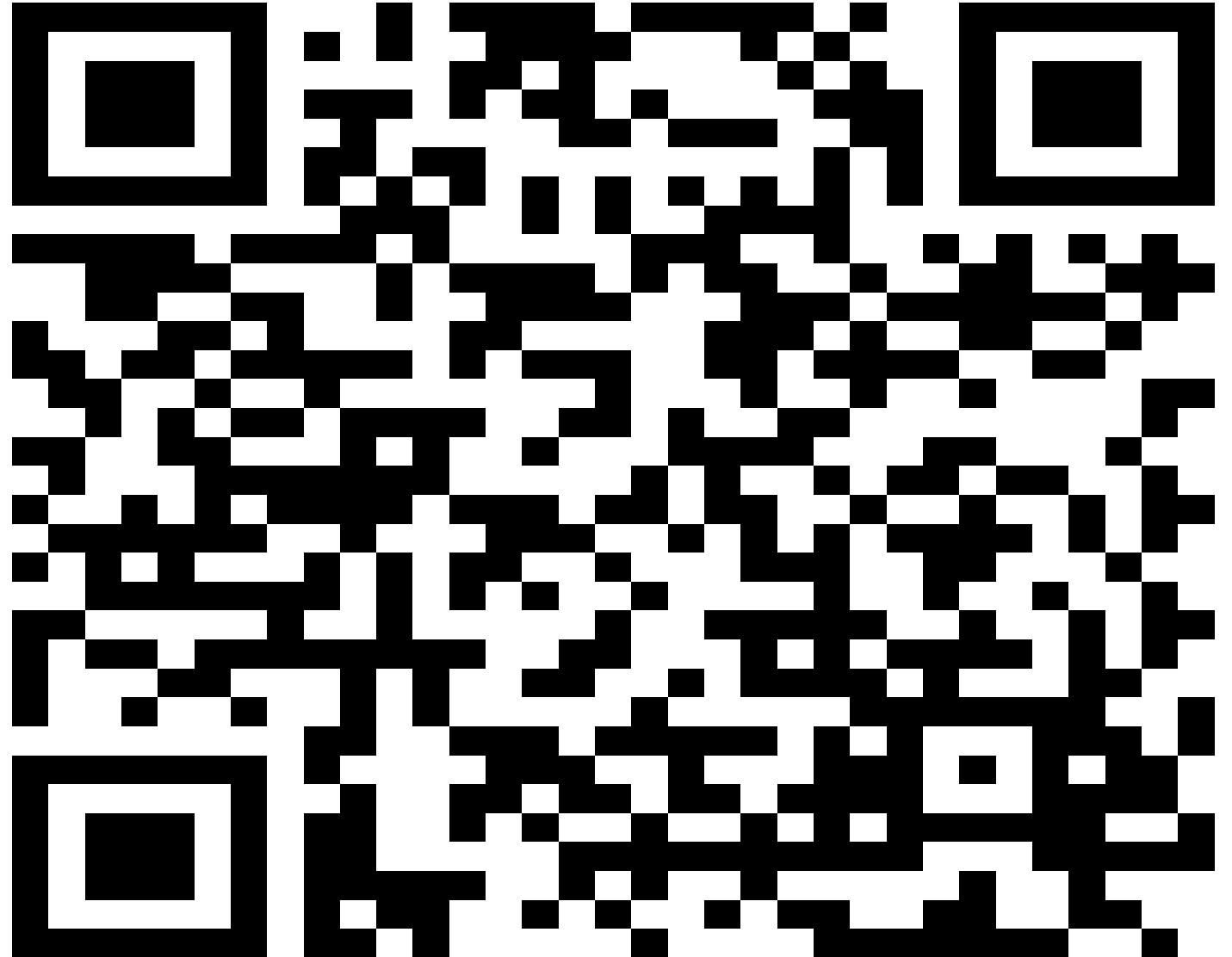
Twitter: @PythonLinks

Skype: clozinski

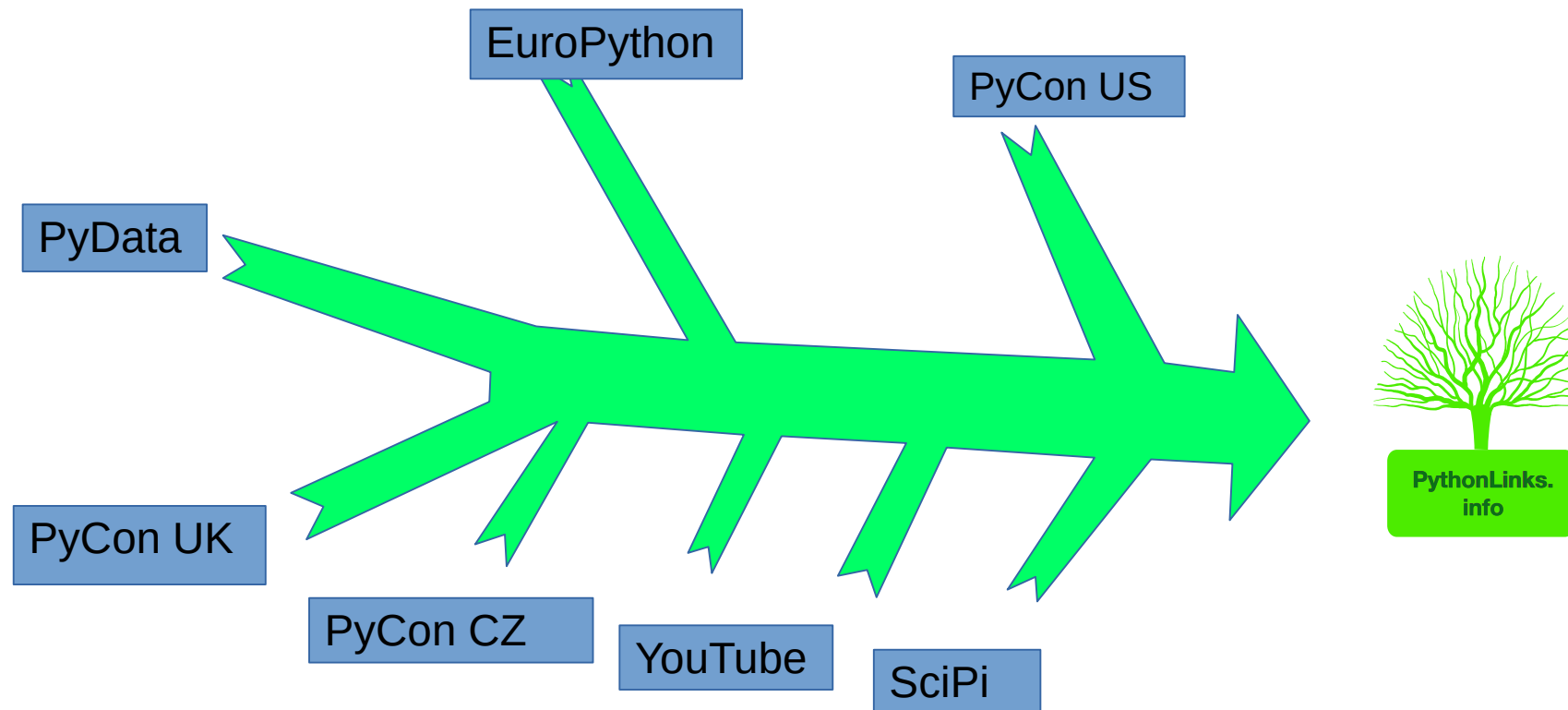
US Phone: +1 (650) 614 1836

EU Phone: +48 12 361 3136

© Christopher Lozi

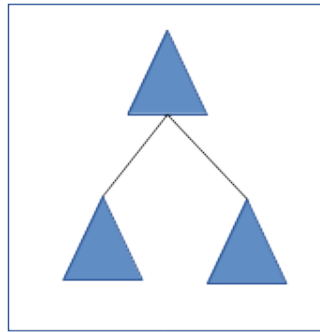


PythonLinks is a Content Aggregation System

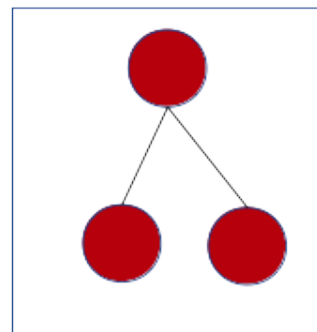


Content Aggregation System

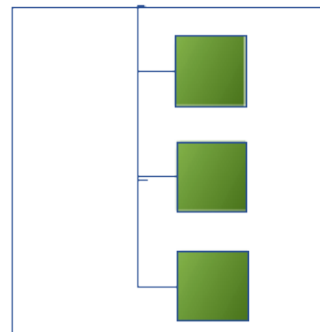
Individual Blogories



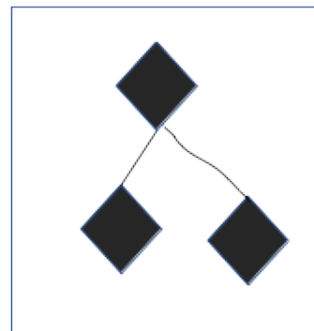
Github Markdown



Web GUI

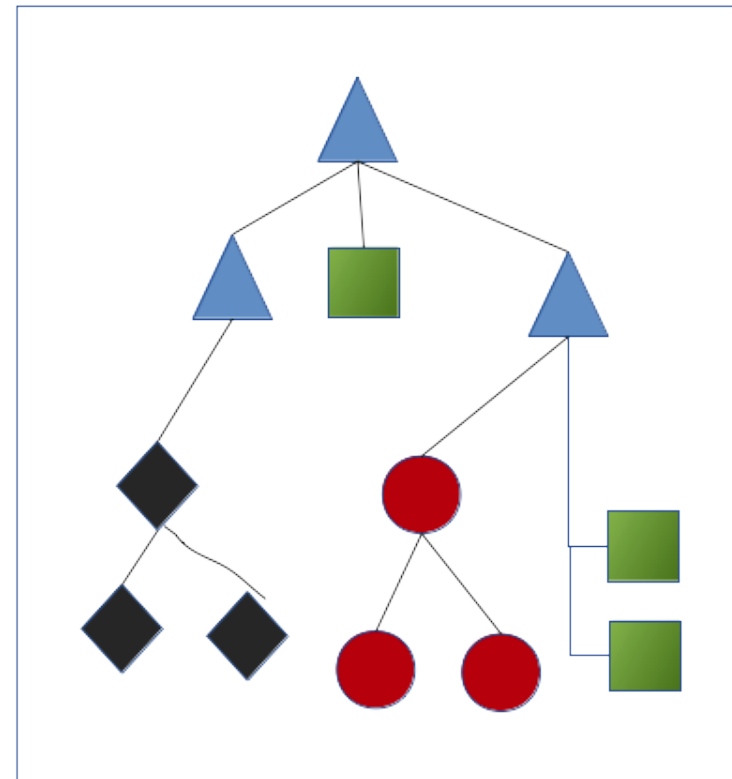


Blog RSS



Plone XML

Group Blogory



Merged Tree

Plone is a Content Management System.

Blogory is also a Content Aggregation System