



Distributed Programming with PYRO (PYthon Remote Objects)

Alfredo Deza
PyCon 2010

*“Simple to get going with,
but is already quite powerful
even in the simplest form”*

Irmen de Jong

- Distributed Object Technology
- Remote Objects behave like local ones
- No external dependencies
- Multi-threaded by default
- Automatic reconnection
- Heavily Tested

- Specialized monitoring
- Distributed Computing
- Testing
- IPC (inter-process communication)

pyro)))

VS



implementation

One master, one slave



Send tasks,
instant
feedback

```
1 pyroloc = 'PYROLOC://127.0.1.1:766/remote'  
2 task = Pyro.core.getProxyForURI(pyroloc)  
3 task.remote()
```


Naming Service

dude where is my object?

What is a

deadlock?

Android calls Nexus



Nexus calls Android



But Android tries to call
Nexus... again?!?!?

Nexus: “The number you have dialed
doesn't exist”

ConnectionClosedError

Validation and encryption



SSL SERVER

```
1 Pyro.config.PYROSSL_CERT="server.pem"  
2 Pyro.config.PYROSSL_KEY="server.key"  
3 daemon = Pyro.core.Daemon(prtcol='PYROSSL')  
4 daemon.setNewConnectionValidator(printCertValidator())
```

SSL CLIENT

```
1 Pyro.config.PYROSSL_CERT="client.pem"
```

handle remote services

the website is down

A horror story:
24 servers,
BASH
and SSH



why **not** to use Pyro?

References:

PYRO:

<http://pyro.sourceforge.net/>

This Presentation and Code:

http://www.alfredodeza.com/pycon_2010.tar.gz



Thanks!

alfredodeza@gmail.com

<http://www.alfredodeza.com>

this presentation was proudly made with Linux, Mercurial and other open source tools