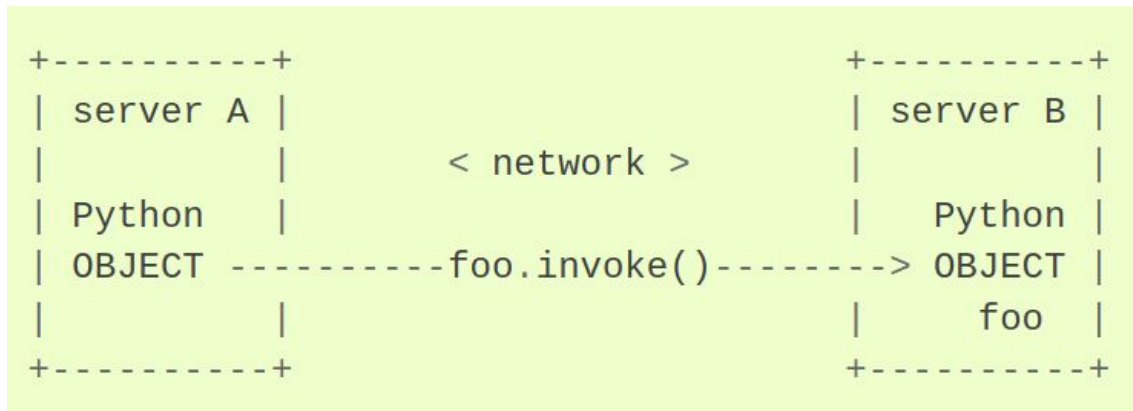# Remote Objects Lab Revision

Pyro

- Pyro is middleware that enables code to call methods on objects even if that object is running on a remote machine; that is, perform Remote Method Invocation (RMI).

```
+----------+                                      +----------+
| server A |                                      | server B |
|          |              < network >             |          |
| Python   |                                      |   Python |
| OBJECT ----------------foo.invoke()--------------> OBJECT |
|          |                                      |     foo  |
+----------+                                      +----------+
```

Pyro name server

- The name server is a utility that provides a phone book for Pyro applications: you use it to look up a "number" by a "name".
- The name in Pyro's case is the logical name of a remote object.
- The number is the exact location where Pyro can contact the object.
- The name server is stated by opening a terminal and entering:

python -m Pyro4.naming

Warehouse

- Warehouse class constructor:

```python
def __init__(self):
    self.contents = ["chair", "bike", "flashlight", "laptop", "couch"]
```

- Warehouse methods:

```python
def list_contents(self):
    return self.contents

def take(self, name, item):
    self.contents.remove(item)
    print("{0} took the {1}.".format(name, item))
```

Register with name server

- The warehouse object registers with the name server

```
Pyro4.Daemon.serveSimple(
        {
                Warehouse: "example.warehouse"
        },
        ns = True)
```

Locate object using name server

- The client locates the warehouse object using the name server

  warehouse = Pyro4.Proxy("PYRONAME:example.warehouse")