MEEC - ASInt
Laboratory 4

Using the basic remote book library as a base, implement a global system (in the LAN of the laboratory) that allows the access of book information available on several servers.

A client should contact the NS (default name server running on SCDEEC network or on sigma) and retrieve a URI for one of the database servers. From this point on, all queries should be forwarded through this server.

The queries to implement are the same as the ones from last week:

- insert book
- search book (using the book identifier)
- List books from a certain author

All operations except book insertion should provide a global list. All servers should be contacted, and the printed information relates to all available servers:

- Search book (using the book identifier): besides the book information, this operation also prints the name of the database (name of the computer)
- List books from a certain author: this operation lists all books available on all databases.

When inserting a book, the information becomes local to the contacted server. There is no replication of information.

Should book identifiers be local or global?

When registering the servers (`ns.register("service_name", uri)`**) should the service name follow a predetermined format?**

In the laboratory computers, when registering an object (**Pyro.core.Daemon**) it is necessary to explicitly state the name of the server: `daemon = Pyro4.Daemon(hots="name_od_machine")`

Name server
https://pythonhosted.org/Pyro4/nameserver.html

The name server (`ns = Pyro4.locateNS()`) is also a pyro service that can be accessed remotely :) This service exports several methods:

- **lookup(name)** Look up the object with the given name and returns the Pyro URI. (it will always return a real PyroURI object, not a string).
- **register(name, URI)** Registers an object with the given name and the given Pyro URI. The URI can be a PyroURI object or just a string.
- **unregister(name)** Removes the object with the given name from the naming database.
- **ping()** Does nothing, just to test if the Name Server is running.
- **list(string)** Returns a ldictionary containg the entreies whose name matches the string