Minimal Core Semantics

- **No application-level semantics**
  Because of legacy modules, and there is no way to predict future semantics.

- **No ordering requirements**
  Improves decoupling, since there is no "right" ordering for every application.

- **Non-atomic**
  Range of stronger requirements provided by higher level tools.

PyBus

A Python Software Bus

Wim T.L.P. Lavrijsen
LBNL

A sofware bus allows for the discovery, installation, configuration, loading, unloading, and run-time replacement of software components, as well as channeling of inter-component communication. The Python programming language encourages a modular design of software written in it, but it offers little or no component functionality. However, the Python language and interpreter have sufficient hooks to implement a thin, integral layer of component support in the form of a Python module. This poster describes such a module, PyBus, with which the concept of a software bus is realised in Python.

http://cern.ch/wlav/pybus

Self-describing objects

- **Builtin into the Python language**
  No further requirements on the bus or on modules/applications.

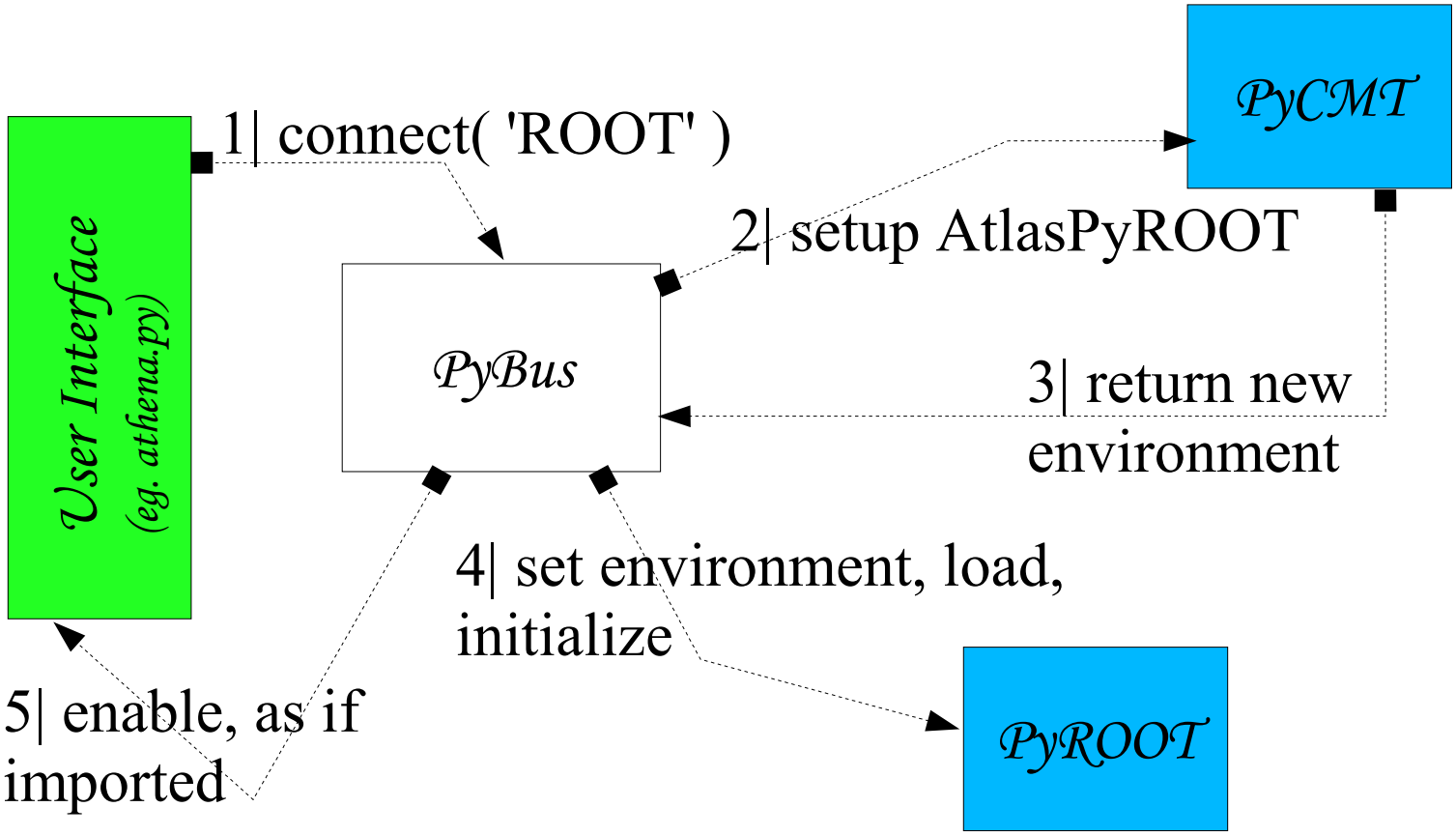
- **Method/Data transformations**
  Convert data types to/from module native types and allow adaptive behaviour.

- **Method/Data transmission**
  Automatic marshalling of both data types and service types.

Dynamic classing

- **Builtin into the Python language**
  No further requirements on the bus or on modules/applications.
- **Create new types on-the-fly**
  Allows old/legacy applications to work with newly defined types.
- **No re-linking necessary**
  System can remain running, while new functionality is added (even written).

An Example

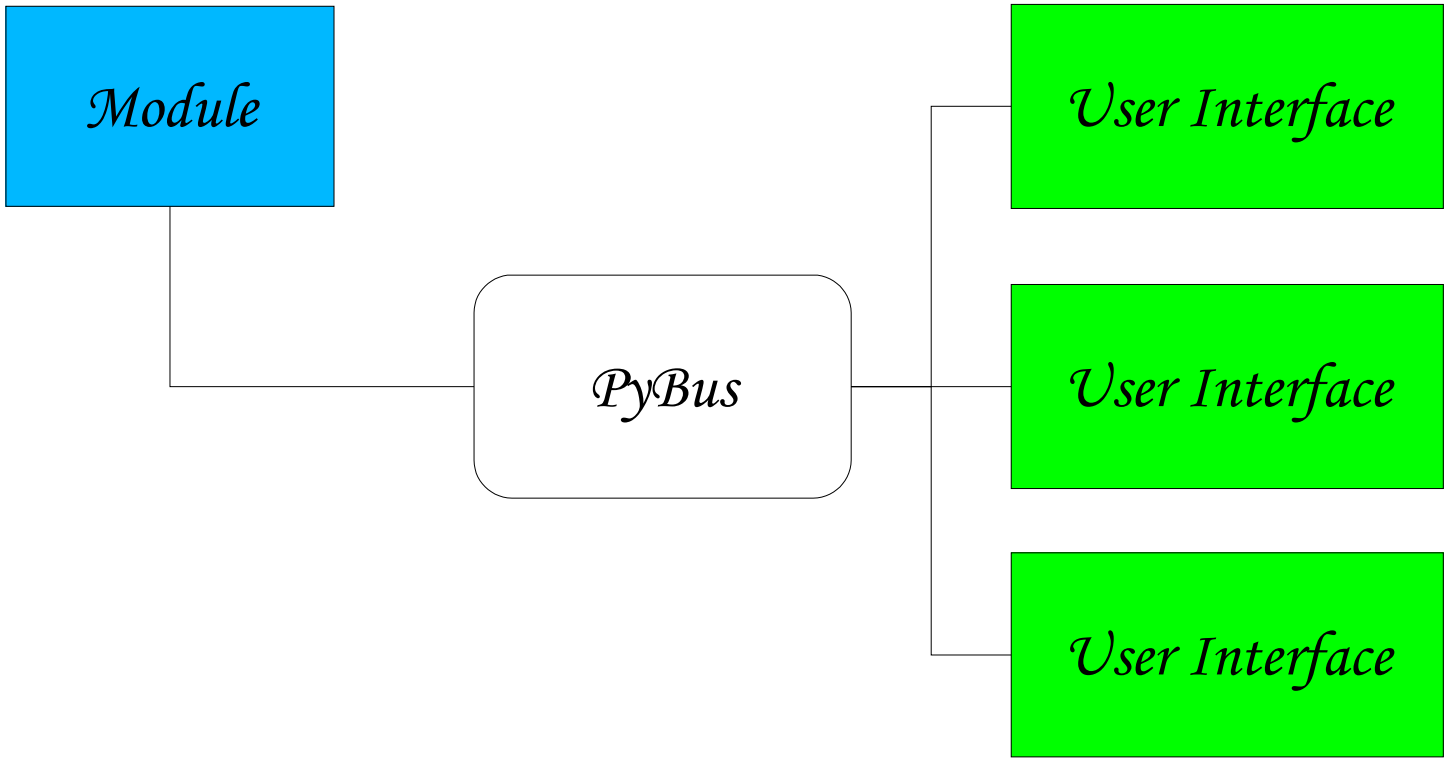


Example: setup PyROOT in an Atlas working environment, using their Code Management Tool. 1) User requests ROOT to be connected. 2) Configuration of PyBus instructs it to use AtlasPyROOT environment to connect ROOT. 3) PyCMT is used to setup the proper environment. 4) PyROOT is loaded into the Python interpreter, pre- and post-configuration of the PyROOT module is handled. 5) The module is “injected” into the user module namespace as if imported.

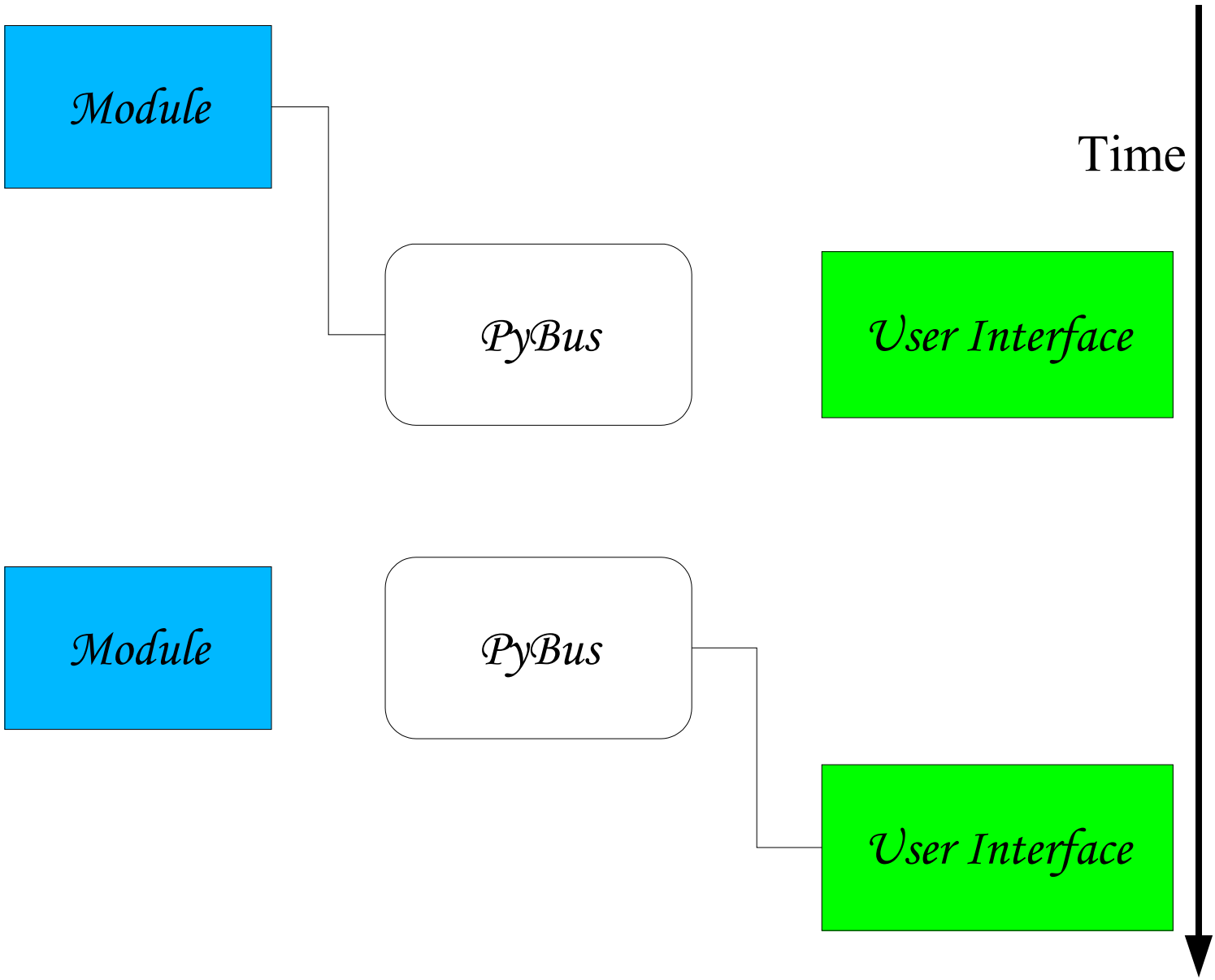
Anonymous communication

- **Address based publish/subscribe**
  String, regex based: no links or references, but need to agree on conventions.
- **Dynamic architecture changes**
  Clients don't care which modules serve them, services don't care who consumes.
- **Policies for communication**
  Confirmation not a priori necessary, but can be implemented (callbacks).

Space Decoupling



Time Decoupling



Synchronization Decoupling