SIG: ROS GUI

TECHNISCHE UNIVERSITÄT DARMSTADT

ROS.org

Wish list

TECHNISCHE UNIVERSITÄT DARMSTADT

- Integrated GUI
  - Framework based on a plugin architecture
    - Permits easy integration of new tools
  - Prerequisite for the following goals
- Increased usability
  - Autosave / restore implemented by the framework
    - Each plugin can contribute information about it's intrinsic state
  - Tool-spanning features like perspectives (known from Eclipse)
- Improved reusability of developed GUI tools
  - Encourages exchange in robotics community

SIM

Choices (to be) made

TECHNISCHE UNIVERSITÄT DARMSTADT

- Which GUI toolkit to use?
  - Fltk (the Fast, Light Toolkit, pronounced “fulltick”)
  - GTK+ (GIMP Toolkit)
  - Qt (pronounced “cute”)
  - wxWidgets (formerly wxWindows)
  - …
- Must have's:
  - Cross platform (Linux, Mac OS X, Windows)
  - License compatible
- Many pros and cons
  - Our proposal is ...

SIM

Qt

TECHNISCHE UNIVERSITÄT DARMSTADT

- Cross platform
  - Supports even mobile platforms
- Very feature rich:
  - UI files for “designing” user interfaces
  - OpenGL, SVG support
  - Support for translating UI
  - Signals / Slots
- Additional sophisticated widgets
  - Qwt (Qt Widgets for Technical Applications)
  - Curve | scatter | spectrogram | contour plots, histogram, sliders, dials, compasses, thermometers, wheels, knobs, …
- Language bindings for C++, Python and more

More Choices

TECHNISCHE UNIVERSITÄT DARMSTADT

- In which language to program?
  - C++ is "lightning" fast
    - But "unlovely" to program and compile during development
  - Python is well suited for rapid prototyping
    - But sometimes "not fast enough"
  - … many other options
    - But these two are the most prominent in the ROS ecosystem
- So, which one to choose?
  - Why not get the best of both worlds?
  - Develop a hybrid application, which enables to use:
    - C++ plugins as well as
    - Python plugins

Hybrid application

TECHNISCHE UNIVERSITÄT DARMSTADT

- Python “main()”
  - Bootstraps Qt using bindings for Python
    - Supports PyQt (GPL) and PySide (LGPL)
    - Since the Qt event loop is running in C++ library there is no negative performance impact
  - Can easily load other Python modules (plugins) at runtime
  - Use C++-Python-bindings to start plugins written in C++
  - Plugins (C++ as well as Python) can be discovered using the ROS package infrastructure
  - For the developer of a plugin:
    - Must not know about the complexity of bindings and the hybrid stuff
    - Just implement a plugin (according to an interface) in one language

What we came up with

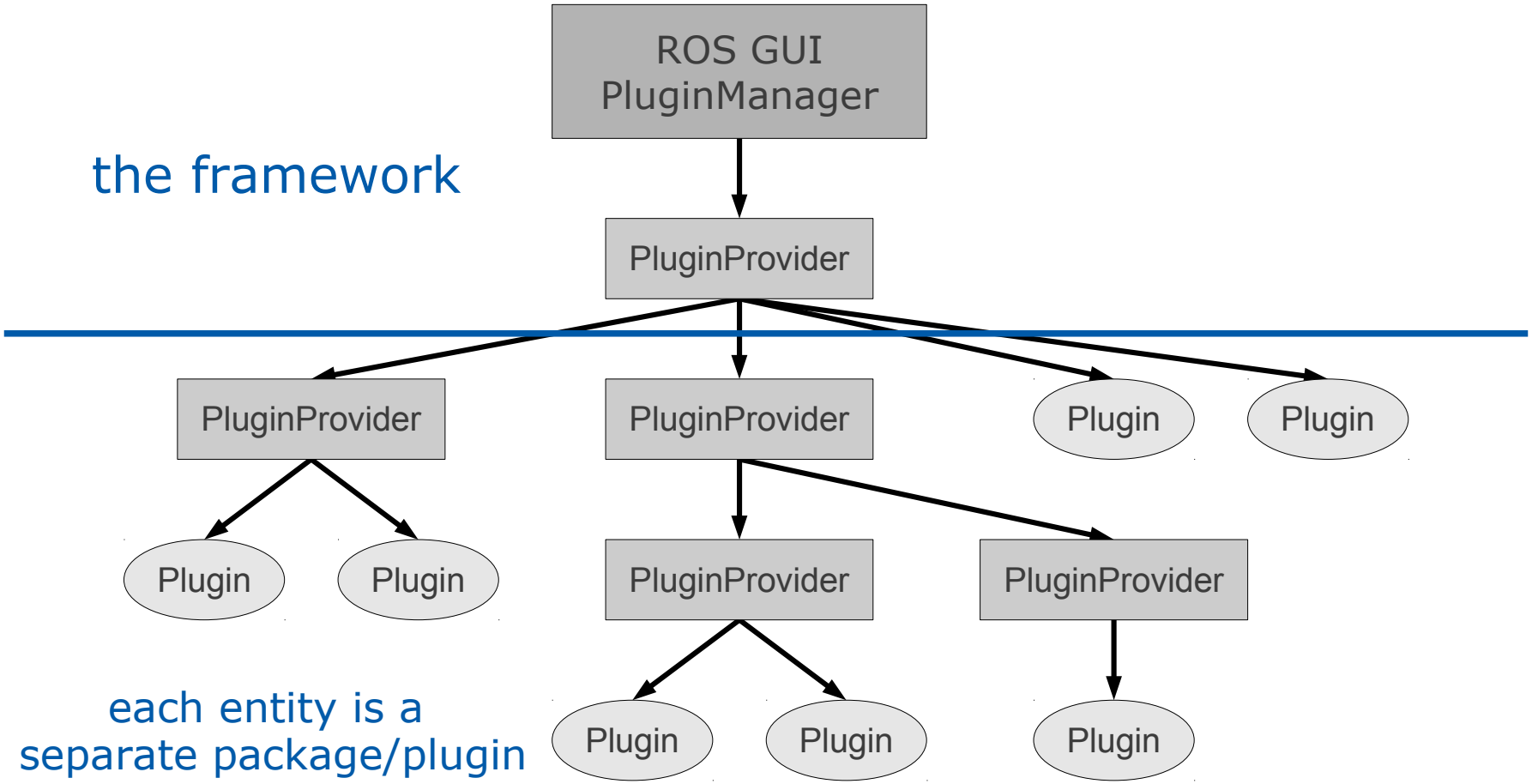
TECHNISCHE UNIVERSITÄT DARMSTADT

- The package rosgui
  - The framework is implemented in Python using Qt (via PySide / PyQt)
  - Implements Autosave / restore functionality, perspective management
  - Uses PluginProviders to discover / load / unload Plugins
    - E.g. RospkgPluginProvider uses the rospkg Python module to integrate ROS packages
    - PluginProviders can be nested
  - Depends only on the Python module rospkg – not on ROS
    - Also applicable for non ROS-based GUI tools
    - Dependency to roscpp/rospy only introduced by specific Plugins / PluginProviders

SIM

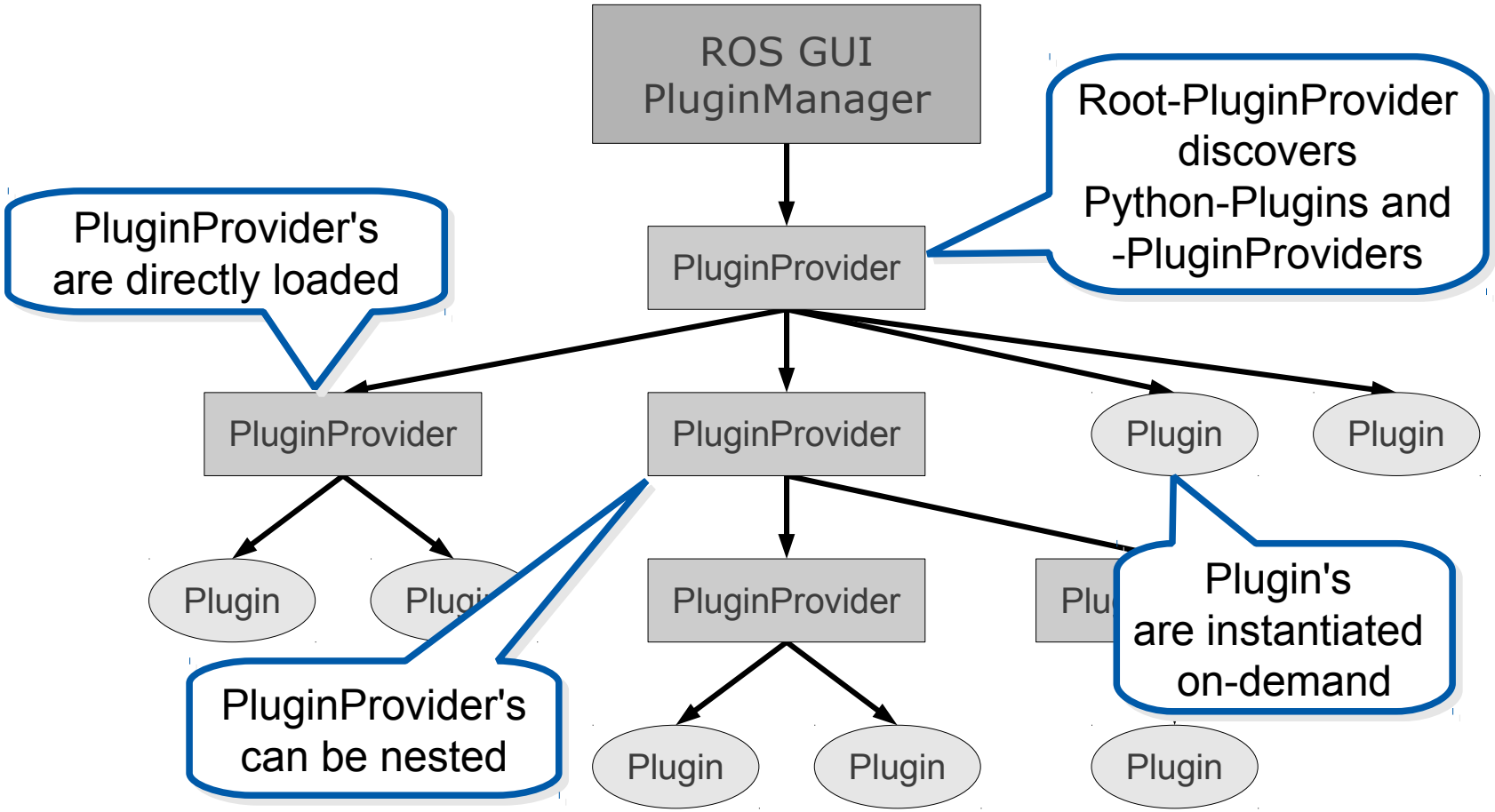
The Framework extended by plugins

TECHNISCHE UNIVERSITÄT DARMSTADT



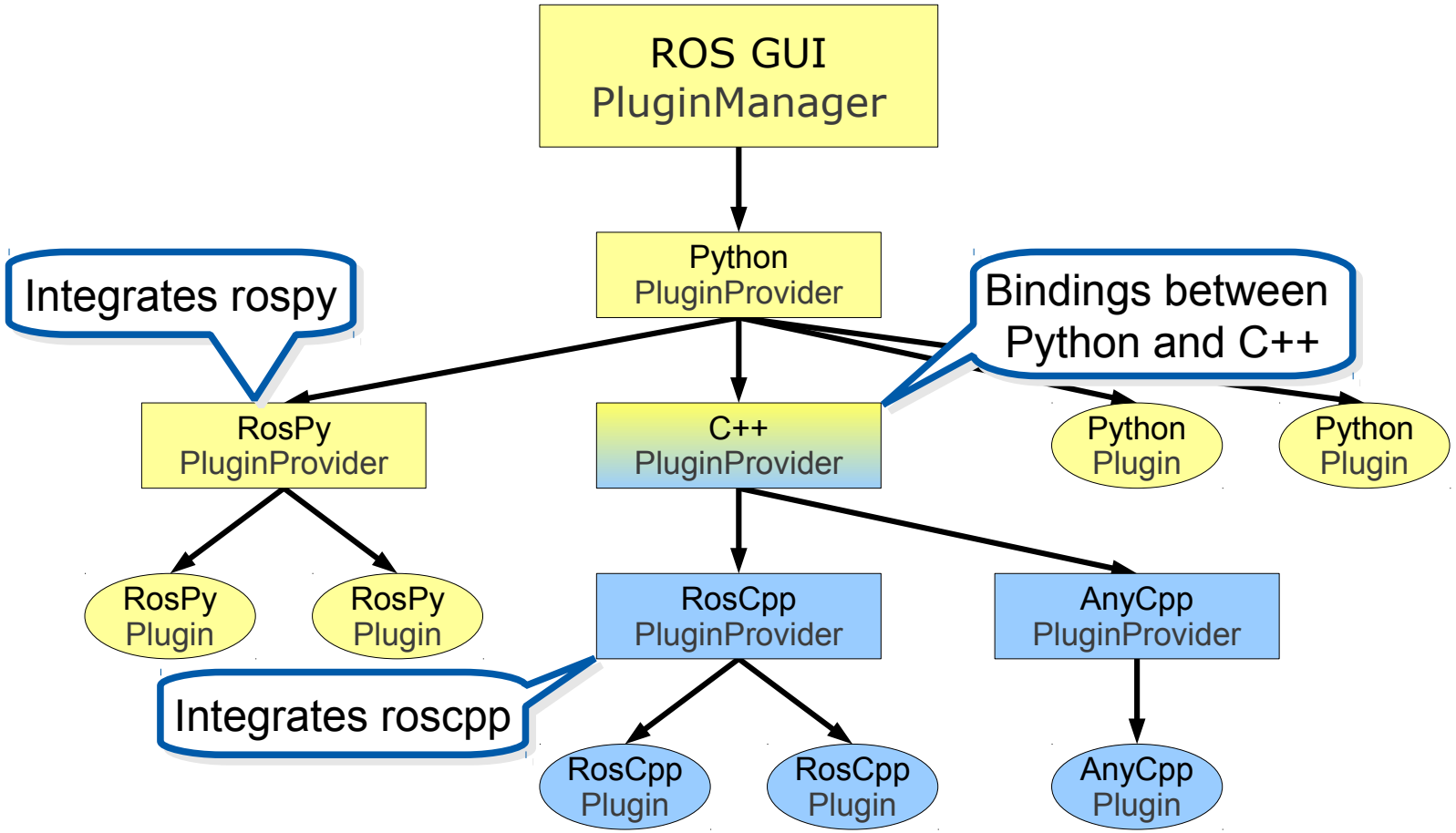
Discovering of plugins

TECHNISCHE UNIVERSITÄT DARMSTADT



Hierarchy of plugin providers

TECHNISCHE UNIVERSITÄT DARMSTADT



Examples for plugins