hosting python web applications

For hosting a python application as a web applications three services are required:

1. Web Server
2. Application Server (WSGI compatible)
3. Python Web Framework

Web Server: redirects a user-supplied url request to a port

- A web server has a few roles.
  - redirect a url to a port number on the local machine.
  - provide an error message when a process does not respond, or too late
  - redirect to another computer: load balancing
- A few web servers are:
  - NGINX
  - Apache httpd (often simply referred to as apache)

Usually the ICT department is responsible for restarting and logging the web server.

For local testing a web server can be left out. Also for users who can handle ugly urls like http://123.456.789.012:9999, a web server is not required to have connect python functionality to the web.

Application Server: launches a python instance and forwards the url-request to it

An application server launches a certain process that can takes the input to a a certain port to handle it. For different languages there are different application servers.

- Java: for Java there is the WAR protocol java as a web server. WAR is a protocol for which numerous implementations exist:
  - Apache tomcat (from same producer as apache httpd web server)
  - JBOSS
- Python: for python there is the WSGI protocol to deploy python as a web server. The Web Server Gateway Interface (WSGI) is a protocol of which numerous implementations exist. Some links are https://www.digitalocean.com/community/tutorials/how-to-deploy-pyramid-based-python-wsgi-web-applications and http://www.fullstackpython.com/wsgi-servers.html.
  - paste:
  - pserve: uses same config as paste
  - gunicorn: linux, for production
  - uwsgi: not recommended, difficult to debug.
  - mod_wsgi: And outdated method was to insert a python application server into the aforementioned web server. For languages like perl and php this is till common, but for python we do not encourage it.
    Usually the ICT department is responsible for restarting and logging the web server in production phase, whereas in development phase it is done by the developers.

Python Web Framework: accepts the url request and returns a response

Inside a running python connected to the web there are several ways to handle the url request. Web Framework exist as small packages or as large packages incl. a database. For web frameworks see this , WikiPedia articale Comparison_of_web_application_frameworks which has a dedicated Python Framework comparison. There are "full-stack" and "micro" frameworks, and some in between (blog).

- Full stack:
  - Django: very complete. Comes with database, views etc.
  - Zope
- In between:
  - Pyramid (developed from pylons). We recommend Pyramid for generating dynamic views on data. Less complete than django, but offers some more room to tweak (such as allowing small pieces of code inside views)
- Micro:
  - Flask: simple
- Specialized:
  - pyWPS: specific python implementation of the OGC WPS standard for geospatial data: interoperability with GIS software. Return object is usually an xml with mime-encoded objects types inside, or a link to source.

Usually the application manager is responsible for restarting and logging the web framework.