Boutiques

Boutiques is a framework designed to publish, integrate and execute command-line applications across platforms in order to improve application integration, reduce redundancy and contribute to computational reproducibility. We can think of it as a facilitator of the FAIR (Findable, Accessible, Interoperable, Re-Usable) principles for software tools.

System description

What is called *descriptors* in Boutiques is just a way for the developers to specify a command line through a JSON file, along with its inputs and outputs. They can be parsed in programming language and also be used to link a container image where the application is installed.

The core component of the descriptor is the command-line template (UNIX shell syntax). It is a single string, containing five value keys, which are placeholders for input and output values. Among other components we also have Input and Output description. We could also define inputs and outputs as configuration files (written in any language). Finally, the command line is constructed from the descriptor at runtime, in an algorithmic fashion.

Invocation Schema

This application-specific JSON object has a single purpose: specify the input values accepted by the application, including the dependencies between them. They can be generated automatically by the command line and stored as an optional property of the Boutiques descriptor.

Workflow support

To achieve framework independency, workflows can be composed from and described as Boutiques descriptors, allowing scalability and reliability.

Containers

Application may be installed in a container image complying with Docker, Singularity or rootfs format.

Resource requirements

Descriptors may contain requirements regarding CPU cores/nodes, RAM or disk storage and walltime. Such properties are called "suggested resources" as the actual ones usually depend on the input, parameters and hardware.

Custom Properties

They may be added to the Boutiques specification without restriction, grouped together in a specific JSON object.

Core tools

Boutiques python package can be installed with "`pip install boutiques`", exposing a command-line utility called `bosh`, providing the below core tools.

- **Validator**: checks conformance of JSON descriptors.
- **Executor**: operates on simulation mode (hypothetical command lines from random values given the descriptor) and launch mode (executes command line from Boutiques descriptor and JSON input file).
- **Invocation** schema handler: creates invocation schema from a Boutiques descriptor (or adds schemas to existing descriptors) and validates input against it.
- **Importer**: converts older descriptors to comply with latest version of the schema.
- **Publisher**: adds an index to the application on NeuroLinks repository.

Some limitations

1. Reproducibility is only partially addressed: containers do not shield against discrepancies arising from different Linux kernel versions or hardware platforms.
2. The application integration bottleneck is moved from integration to validation.
3. Containerized applications are not secure, with Docker being the riskier one.

Conclusion

With Boutiques, developers can integrate their applications once and execute them in several platforms. Boutiques removes the technological dependency to a particular platform and facilitates application migration.