Dependency Injection

Real world example

As an autonomous work- or craftsman, how to acquire contracts?

Would you burgle arbitrary houses in order to remodel, say, the occupant's kitchen the way **you** would like it to be?

Rather not! Most likely you will be contacted by occupants who want their kitchen to be remodeled or you may ask them whether they want it to be remodeled.
Anyway, they will tell you how **they** want it to be remodeled.

The only thing you have to know is how the customer's remodeling is done by means of tools to be used as well as craftsmanship.

Dependency Injection

Findings from this example: The principle

The customer is always right! As a service provider you will have to kindly ask her or him what she or he would like to be done.

You won't remodel somebody's kitchen without asking her or him beforehand!

Only the particular customer can really tell you what she or he wants to be done (first-hand or reference information).

Your key question is “may I kindly ask you to provide me with your means (in terms of knowledge) **I depend on** in order to do my work?”

Dependency (Injection) Container

Understanding the virtual concept

For an easier understanding, let's forget about the “injection”;
instead, let's add the concept of a container holding dependencies.

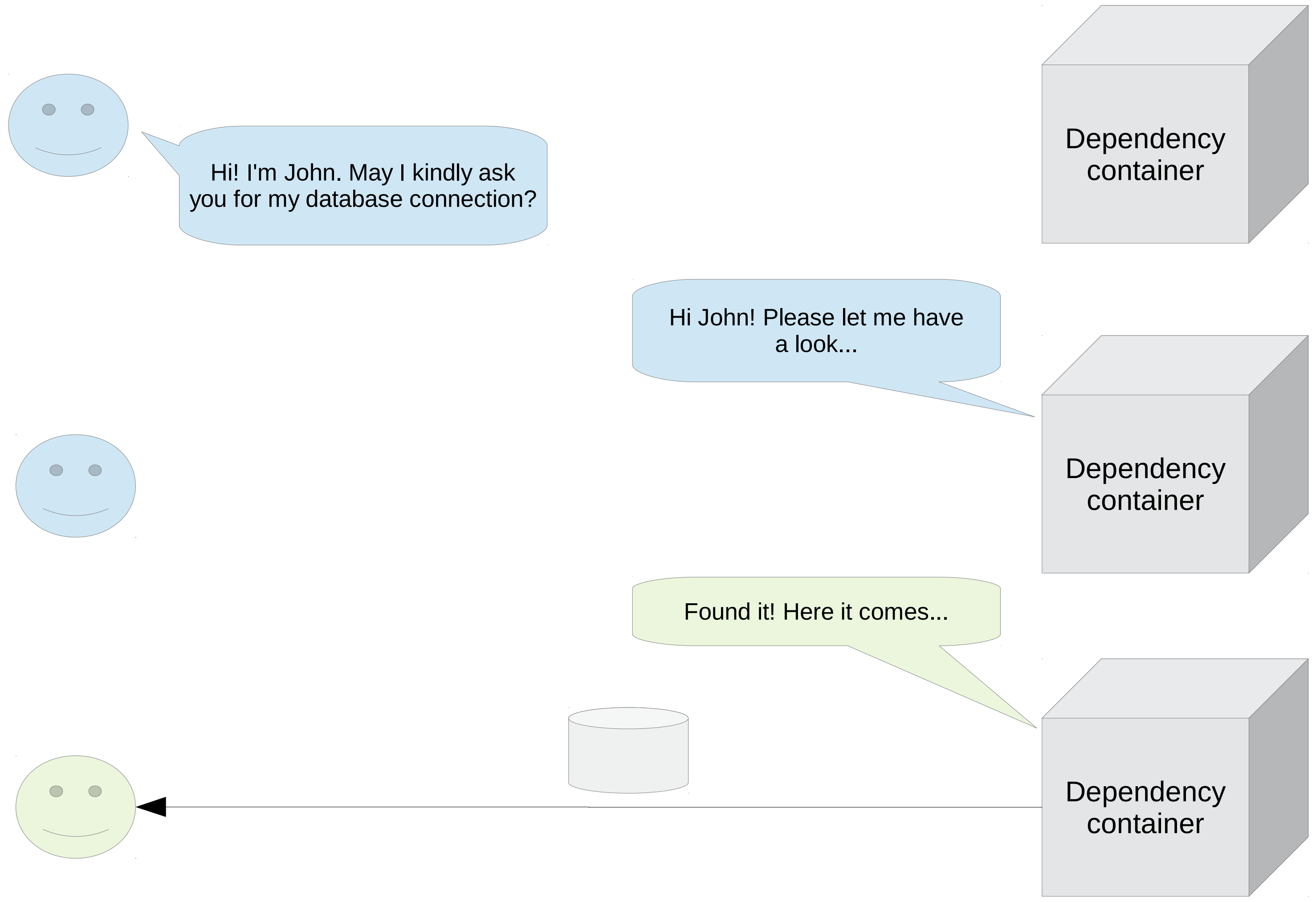
The dependency container is the only instance (in terms of a reference)
that knows about the insights of the dependencies.

So it's recommended to use the dependency container as a dictionary
that white-lists valid dependencies.

From outside the container, you cannot change dependencies,
but only request the same.

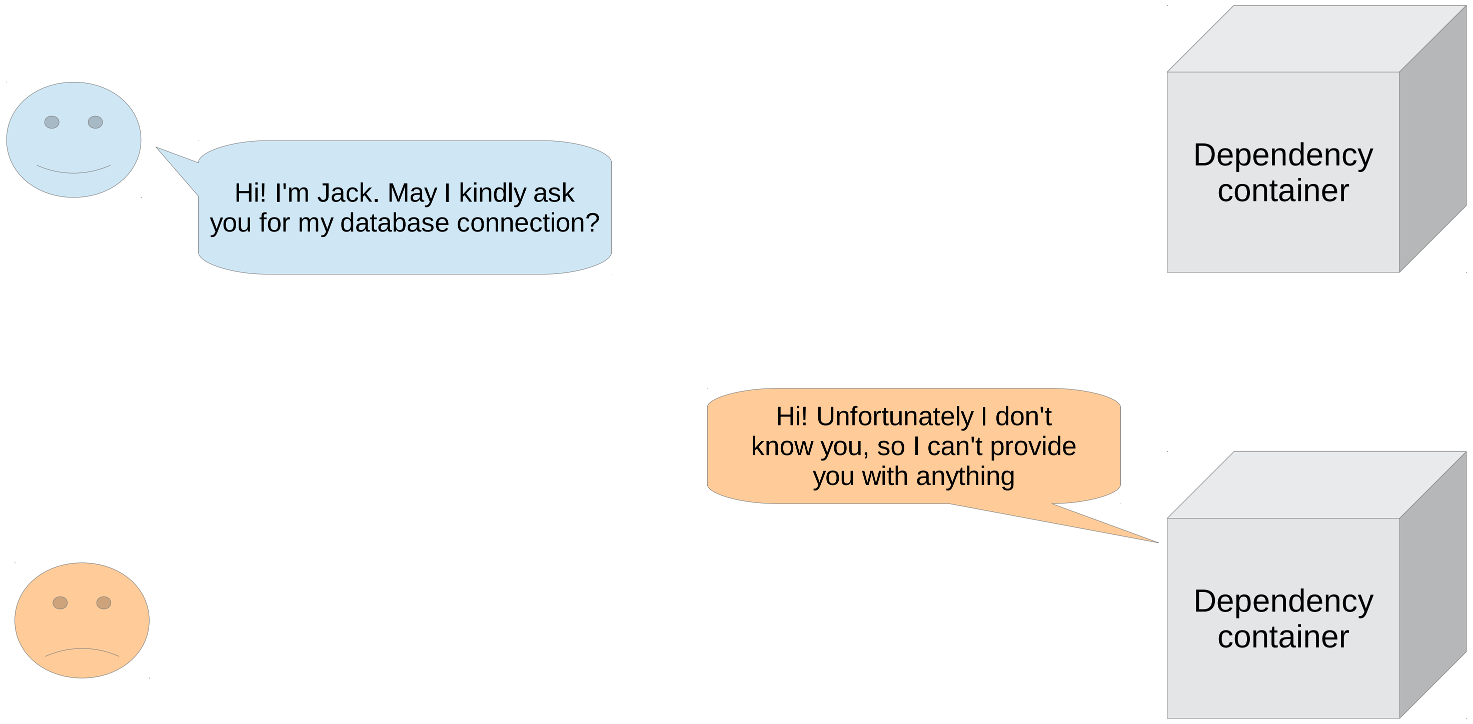
Dependency Container

A graphical representation


Hi! I'm John. May I kindly ask you for my database connection?
Dependency container
Hi John! Please let me have a look...
Dependency container
Found it! Here it comes...
Dependency container

Dependency Container

Use it as a reference


Hi! I'm Jack. May I kindly ask you for my database connection?
Dependency container
Hi! Unfortunately I don't know you, so I can't provide you with anything
Dependency container

Dependency Injection

Summary: Concept or implementation?

Dependency injection (DI) may be implemented
using any programming language.

So, in fact, DI is a **concept** rather than an implementation.

The facts that e.g. Java inherently implements DI or that there
are DI container frameworks may distract from this finding.

As a concept, DI applies the pattern of the "**inversion of control**".
This means that there is no omnipotent craftsman.

As a registry or (for the sake of security) a read-only dictionary,
a dependency container applies the pattern of a "**singleton**".

Dependency Injection

Summary: Concept or implementation? (cont.)

If, on execution, a function or class is provided with its dependencies without prior request, this is referred to as an **injection**.

Dependency (Injection) Container

Btw: What are dependencies?

Dependencies are any kind of resources a requester may depend on.

Examples may be
- an application white-list (registry or dictionary)
- references to session objects
- configuration settings
- references to database connection objects
- data descriptions
- language modules

...

Dependency (Injection) Container

Summary: Advantages

Security

The inversion of control inhibits undesired results and thus behavior as a requester cannot alter the container's internal logic.

This is especially true when being used in typical web application environments where the container is a plain back-end implementation not reachable by the front-end (by different run-time environments and encapsulation).

Clarity, trust and maintainability

The container's nature as a singleton makes it the only source of trust for dependencies, so a dependency is either always wrong or always right regardless of the requester.

Maintaining or changing a dependency effects all its requesters at once.

Dependency (Injection) Container

Summary: Possible disadvantages

Design effort
When designing your own DI container, working out a maintainable and effective design may be expensive.

Implementation effort
Implementing your own DI container of course bears the respective effort.

No golden hammer
Using a DI container will mainly increase the maintainability of your code in terms of refraining you from bypassing your design, but it won't necessarily refrain others (like an attacker) from doing so (e.g. by injecting SQL etc.).

Dependency (Injection) Container

Implementation: Pseudo code example

```
For a pseudo code example, see http://reichartonline.de/?page_id=533
```

By using this structure, you can put any dependency into your container and easily request it.

If you don't want to seal your container, you can offer an interface to register dependencies from outside.