Side by Side Deployment

Kevin Jones

@kevinrjones

Overview

- Might want to load different versions of the same class
  - IDE extensions require different verions of XML parsers
- Classloader provides isolation

Class.forName

- Three parameter version of Class.forName
  - Class is loaded by the classloader in the 3rd paramter

```
ClassLoader ucl = new URLClassLoader(...);
Class.forName("com.pluralsight.Quote", true, ucl);
```

Calling Down the Loader Hierarchy

```
Class rocketCls= Class.forName("Rocket", true, ucl);
Rocket r = (Rocket) rocketCls.newInstance();
```

Incorrect. Rocket should not be
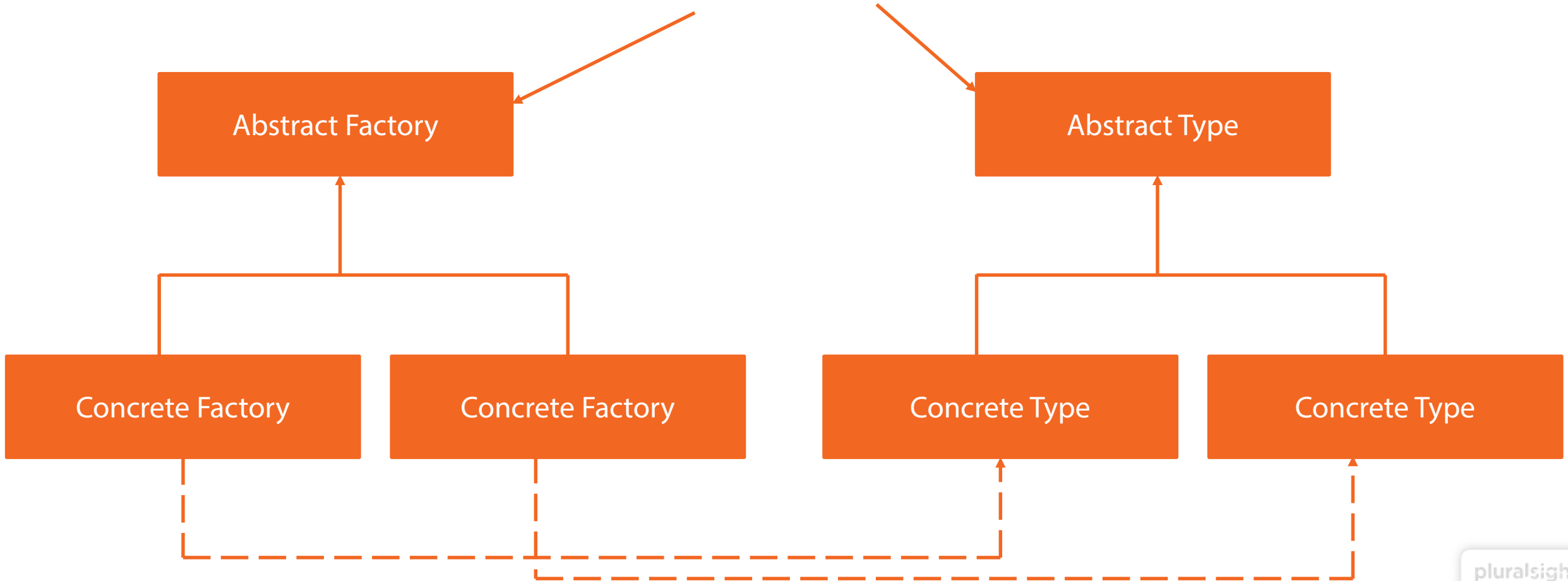visible
to caller's loader

Correct. Rocket should
be visible to
URLClassLoader

```
Class rocketCls= Class.forName("Rocket", true, ucl);
RocketItf r = (RocketItf) rocketCls.newInstance();
```

Factory Pattern

- Constructor pattern

Summary

- Use interfaces when dynamically loading classes
  - Classes you are loading cannot be on the classpath
  - Still need a way of calling methods on those classes
- Can create flexible factories
  - Use classloader to load factory dynamically