Introduction

It is not always desirable to package application configuration with the application. For example, some configurations need to be secured, are dynamic or are based on the running environment. As a consequence, these configurations need to be stored externally. This is even more prominent in the microservice programming model, where the runtime environment for a microservice can change and so must its configuration. The twelve-factor App states good practices of delivering cloud applications, and the third item there is the configuration.

A number of open-source projects such as Archaius, Apache Commons Configuration, Tamaya and DeltaSpike are trying to deal with application configuration, be it static, internal, dynamic or external. But there is no standard API to unify these efforts. This makes application portability between providers very difficult.

This proposal addresses the configuration requirements for all Jakarta EE programming models, so all Java applications not only microservices should benefit from this effort.

Motivation

This specification will allow several Jakarta specifications such as JPA, JMS, and NoSQL to follow the twelve factors guidelines and have a centralized configuration as CDI does with an injection of dependency and context.

This Configuration API will allow microservices to run in different environments seamlessly. The proposed Configuration should solve the following three issues:

1. Allow configuration files defined in different locations - inside and outside applications
2. Provide a way to properly order the property files in a deterministic way
3. Cope with dynamic changes in configuration files

Influences and History

JSR 382:

https://github.com/eclipse/ConfigJSR
MicroProfile Configuration:
https://github.com/eclipse/microprofile-config
Helidon Configuration:
https://github.com/oracle/helidon/tree/master/config