Chinook Guide: Auto-Configuration
Author: Stephen Montgomery
Date: Feb. 5th, 2004

Description

JXTA v2.2.1 contains new functionality designed to aid in the configuration of "peers". Previous versions of JXTA have relied solely on the `DefaultConfigurator` and the corresponding AWT configuration utility. The goal of the JXTA community is to move away from client-facilitated configuration to fully dynamic configuration. This transition is being achieved through incremental additions to the `ext.config` package. In the meantime, Chinook has implemented elements of this transitioning technology to aid in the auto-configuration of its client and server peers. This document outlines the structure of the Chinook auto configuration package.

Bootstrapping JXTA

The bootstrapping process in JXTA can be described as the automatic determination of a peer's configuration. Several elements of a peer's configuration need to be bootstrapped in Chinook:

- Transports: The HTTP, TCP and Multicast properties of a particular peer
- Rendezvous status: Whether the particular peer should become a rendezvous, can become a rendezvous, and what rendezvous' this peer should be aware of at start-up
- Relay status: Whether the particular peer should become a relay or use a relay, and what relay's this peer should be aware of at start-up
- Security: The peer name, ID, and password need to be maintained from independent of executable instance and in an informative way on the network.

At this point, this may not be a complete list.

Package ca.bcgsc.chinook.p2p.configuration

The goal of this package is to load bootstrapping information from the appropriate XML file, specific to either the client or the server. This XML file contains the configuration information to create a new instance of the `PlatformConfig`. When values of this XML file are changed, the `PlatformConfig` file at `JXTA_HOME` must be deleted. These files are loaded via the ClassLoader and therefore must always have the same name and be in the Chinook `CLASS_PATH`.

Location of Bootstrapping files:

1) Server:
    a. `resources/server-config.xml`
2) Client
    a. `resources/client-config.xml`

The format of the bootstrapping configuration files in identical.

```
<client-config>
      <configurator_class>ca.bcgsc.chinook.p2p.configuration.ext.ClientConfigurator</con
figurator_class>
<!-- Use this configurator class below, instead of the one above to get the default JXTA
configurator -->
```

```
<!--  <configurator_class>default</configurator_class>  -->
      <security>
            <name></name>
            <password></password>
            <autogenerate>true</autogenerate>
      </security>

      <transport>
            <type>tcp</type>
            <address>1.2.3.4</address>
            <port>9701</port>
            <multicast_address>224.0.1.85</multicast_address>
            <multicast_port>1234</multicast_port>
            <multicast_size>16384</multicast_size>
      </transport>

      <transport>
            <type>http</type>
            <port>9700</port>
            <proxy>http://usr:pwd@myProxy.myDomain:8080</proxy>
      </transport>

      <rendezvous>
            <addr>http://coast.bcgsc.bc.ca:9700</addr>
            <addr>tcp://coast.bcgsc.bc.ca:9701</addr>
            <addr>http://localhost:9700</addr>
            <addr>tcp://localhost:9701</addr>
            <activate_as_rendezvous>true</activate_as_rendezvous>
            <can_become_rendezvous>true</can_become_rendezvous>
      </rendezvous>

      <relay>
            <addr></addr>
            <addr></addr>
            <activate_as_relay>false</activate_as_relay>
            <can_become_relay>false</can_become_relay>
            <use_relay>true</use_relay>
      </relay>

</client-config>
```

Description of Bootstrapping Configuration files

1. `configurator_class` tag
    a. This is either the server or the client configurator class. This tag is redundant in this architecture and will likely be deprecated. Currently, it needs to be set to the appropriate class.
2. `security` tag
    a. The user `name` and `password` for this node. If neither `password` or `name` is set AND `autogenerate` is set to true, these will be determined for this peer. The default is to set the `name` and `password` equal to the system user name.
3. `transport` tag
    a. The available transports. If these aren't set to match your network interface, they will be reconfigured from the bootstrapping values to valid transports. The two types of transport that are available are `HTTP` or `TCP`.

4. `rendezvous` tag
    a. This tag contains default rendezvous locations. These locations are augmented with those from the SUN site.
5. `relay` tag
    a. This tag contains default rendezvous locations. These locations are augmented with those from the SUN site.

UML Diagrams

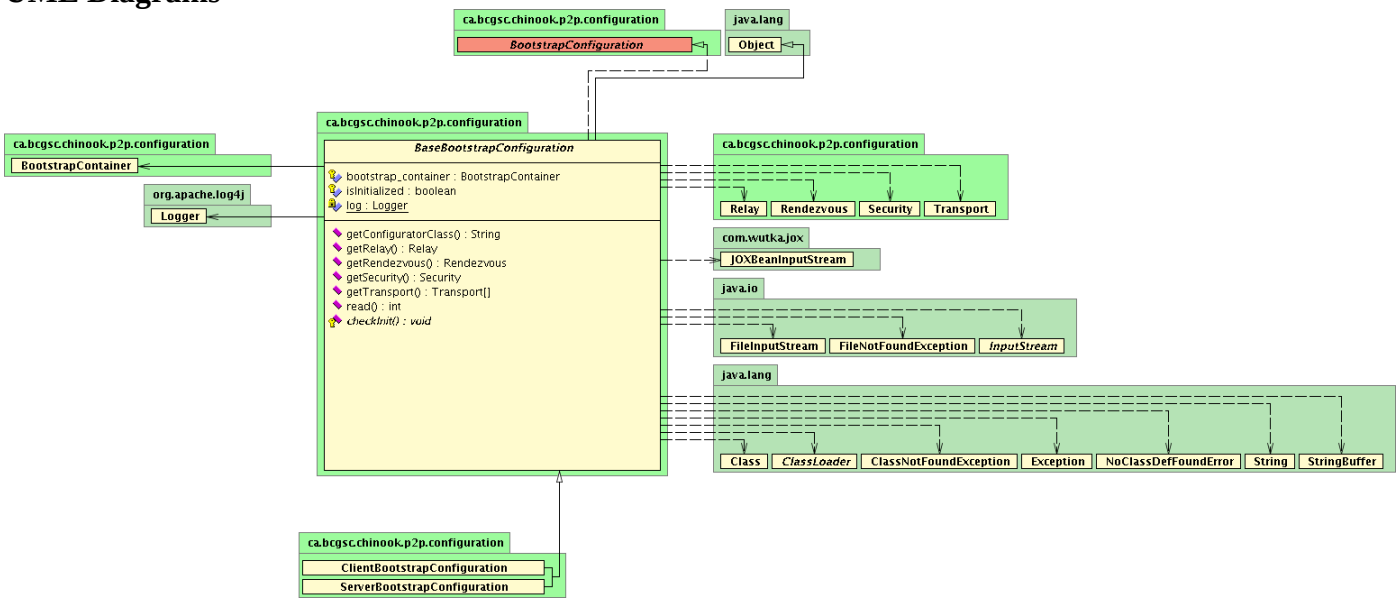

Figure 1: BaseBootstrapConfiguration class

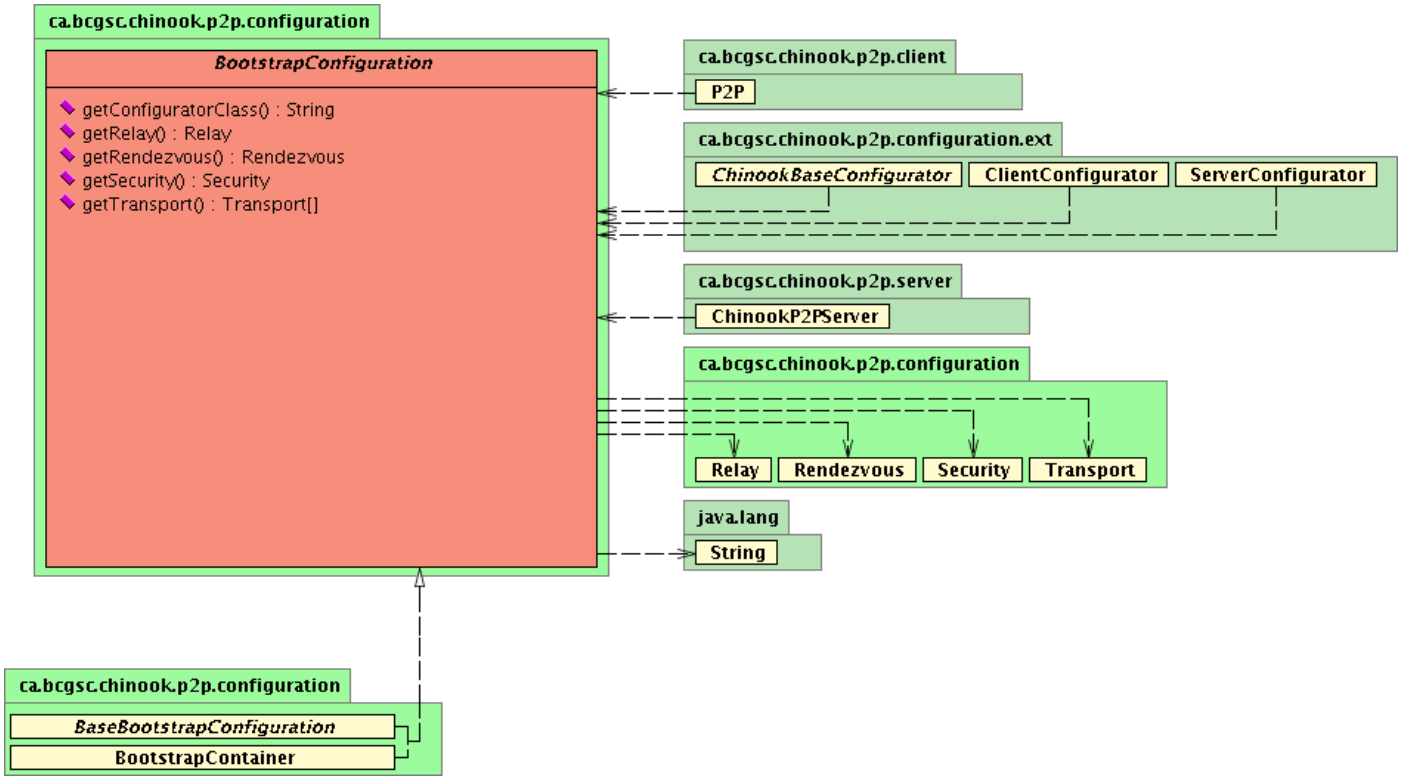

Figure 2: BootstrapConfiguration interface

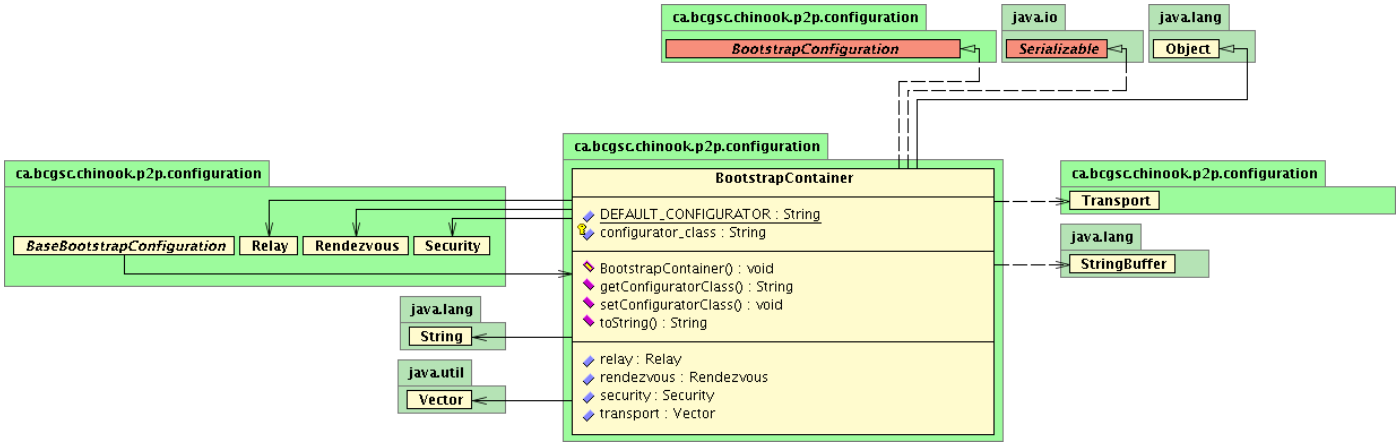

Figure 3: BootstrapContainer class

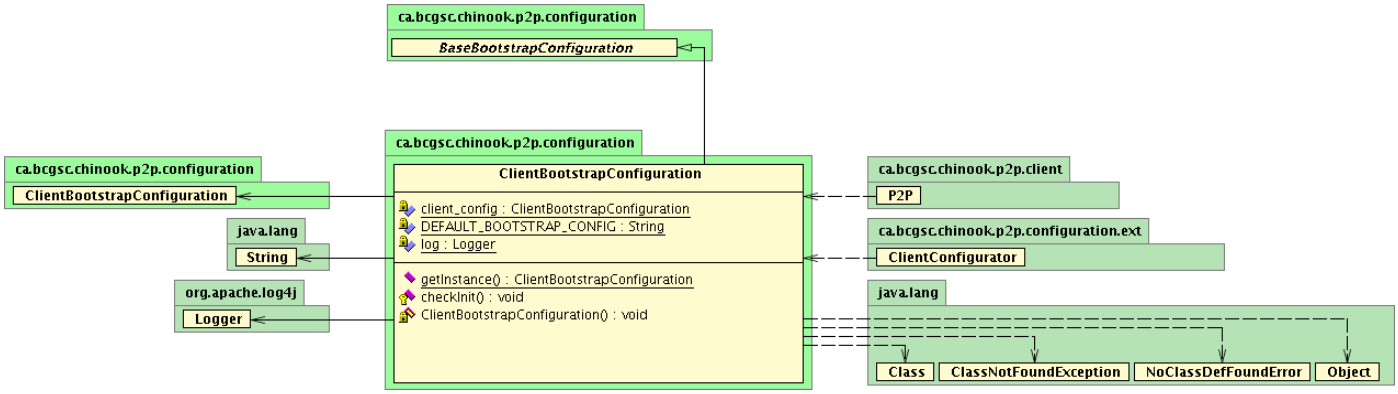

Figure 4:ClientBootstrapContainer class

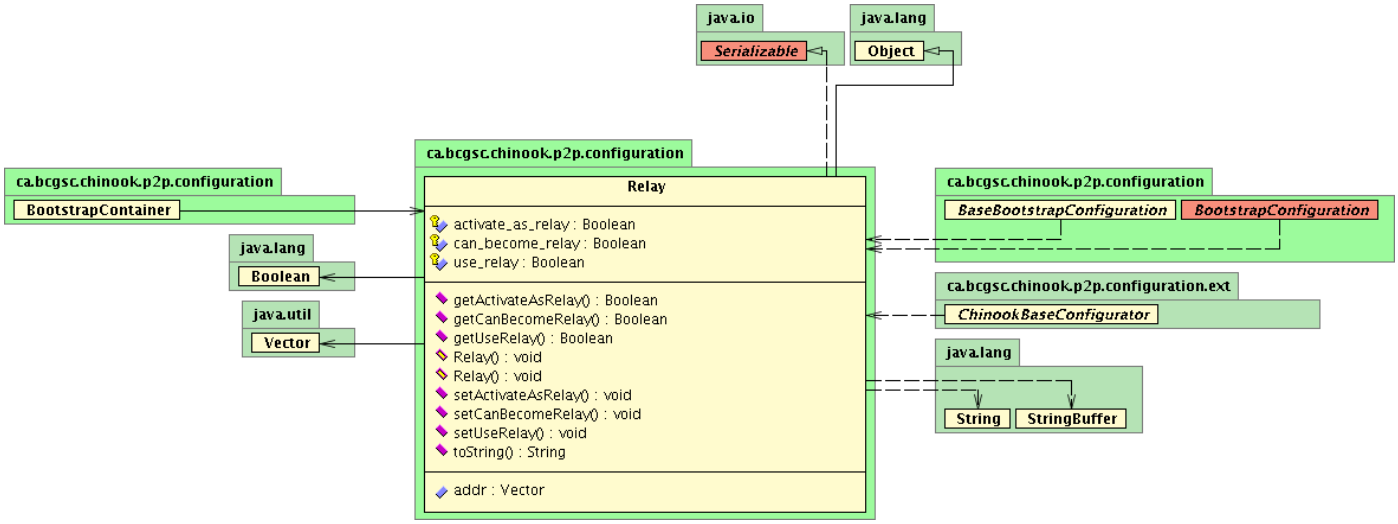

Figure 5: Relay class

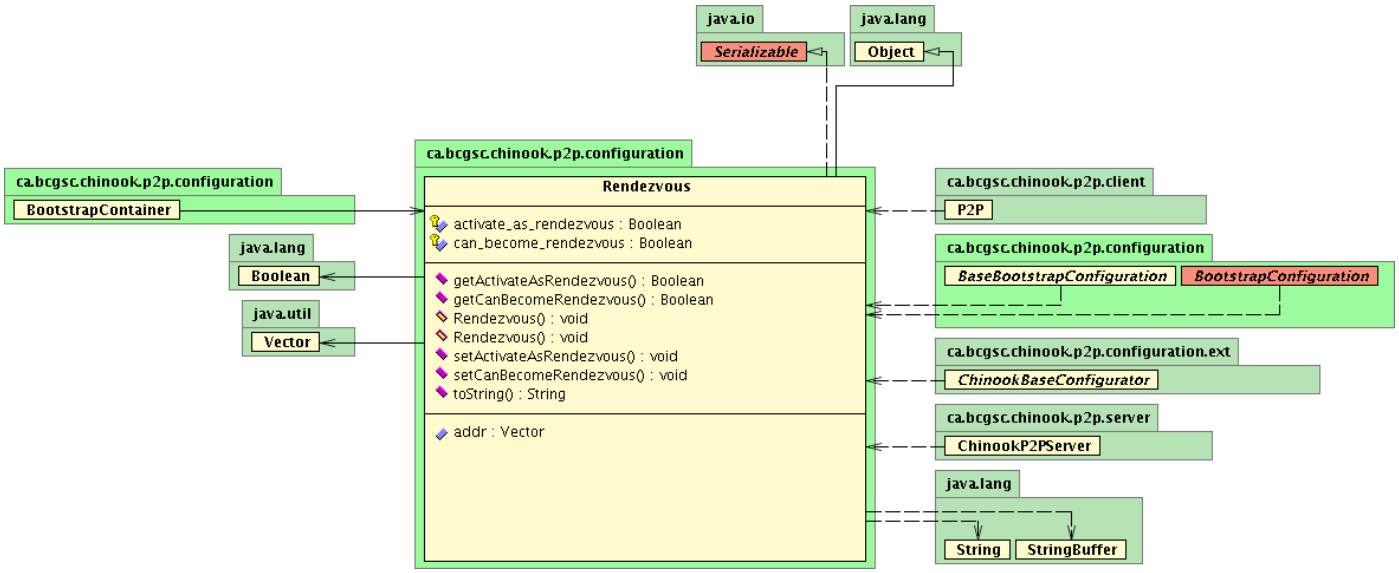

Figure 6: Rendezvous class

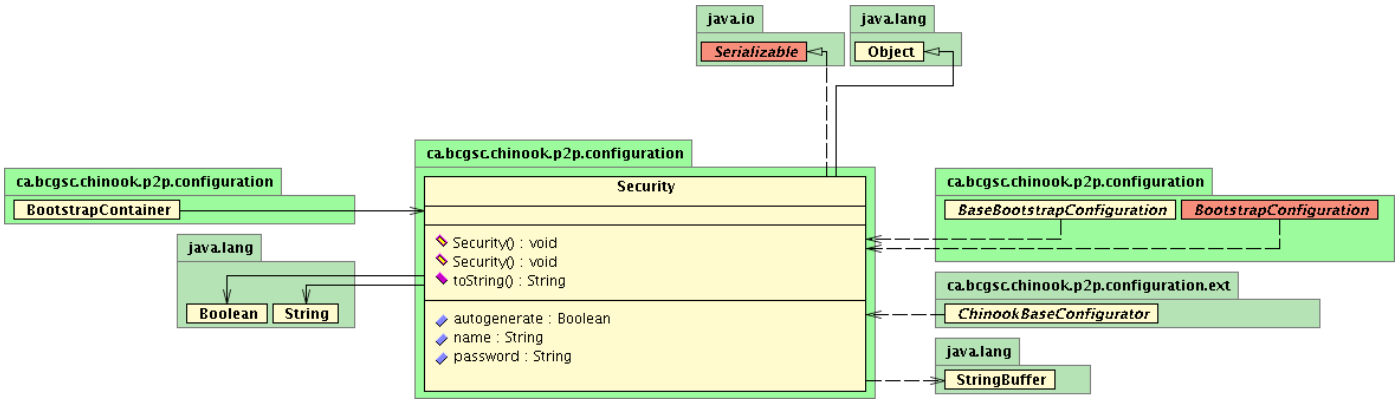

Figure 7: Security class

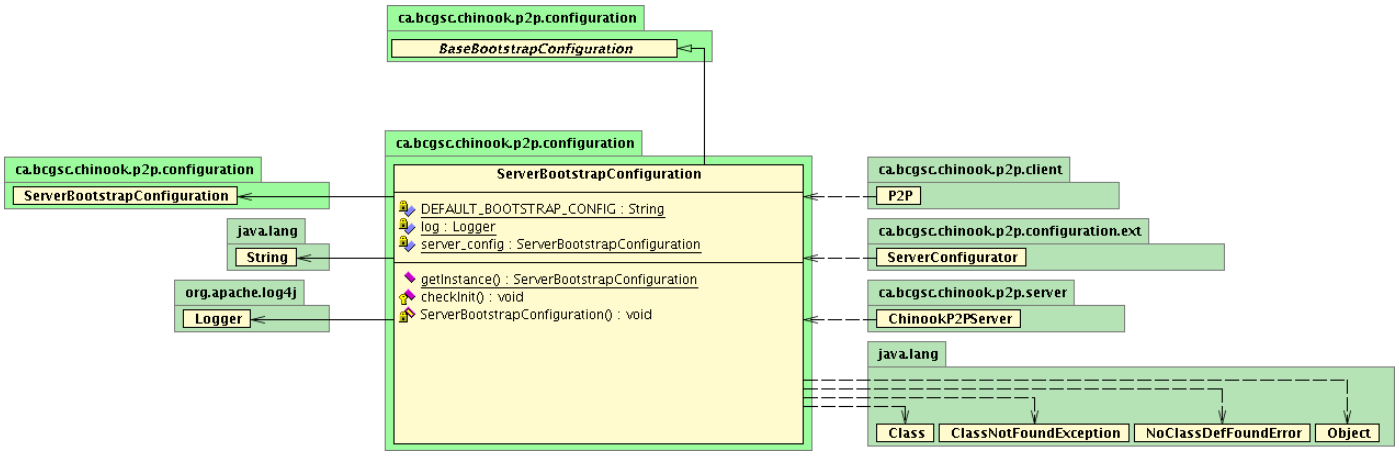

Figure 8: ServerBootstrapConfiguration class

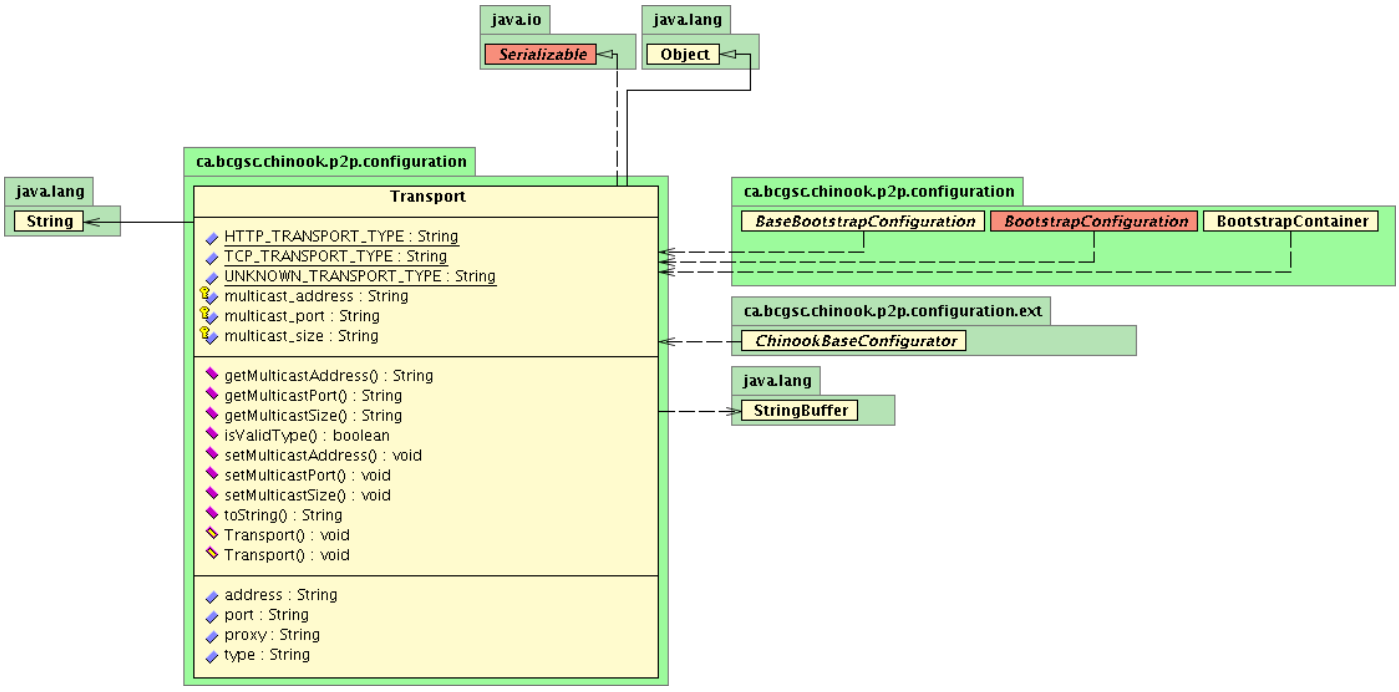

Figure 9: Transport class

Package ca.bcgsc.chinook.p2p.configuration.ext

The goal of this package is to take the bootstrapping data and create a new JXTA Configurator (`ext.config` package). The majority of the service code is in the `ChinookBaseConfigurator` class.

UML Diagrams

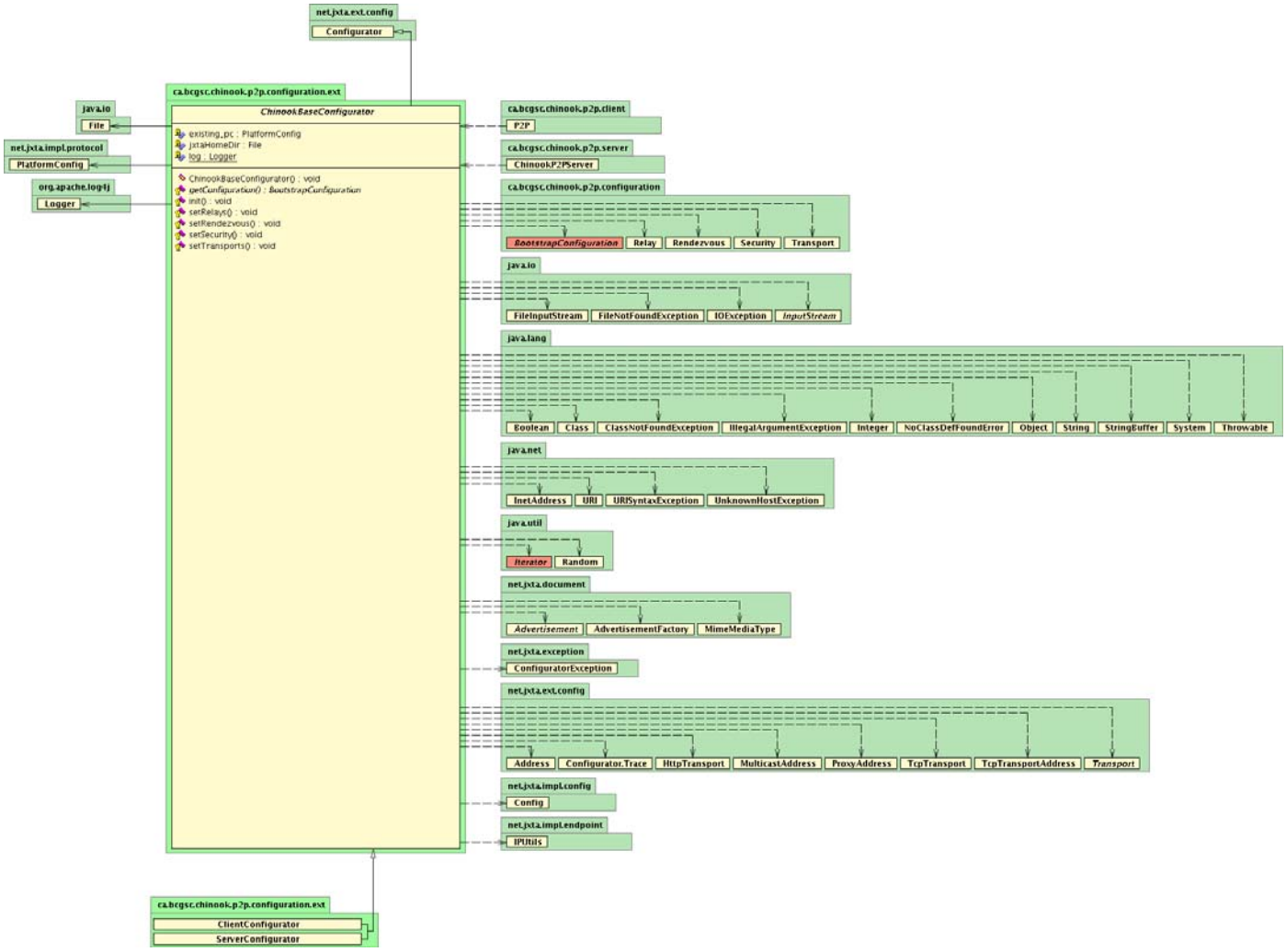

Figure 10: ChinookBaseConfigurator class

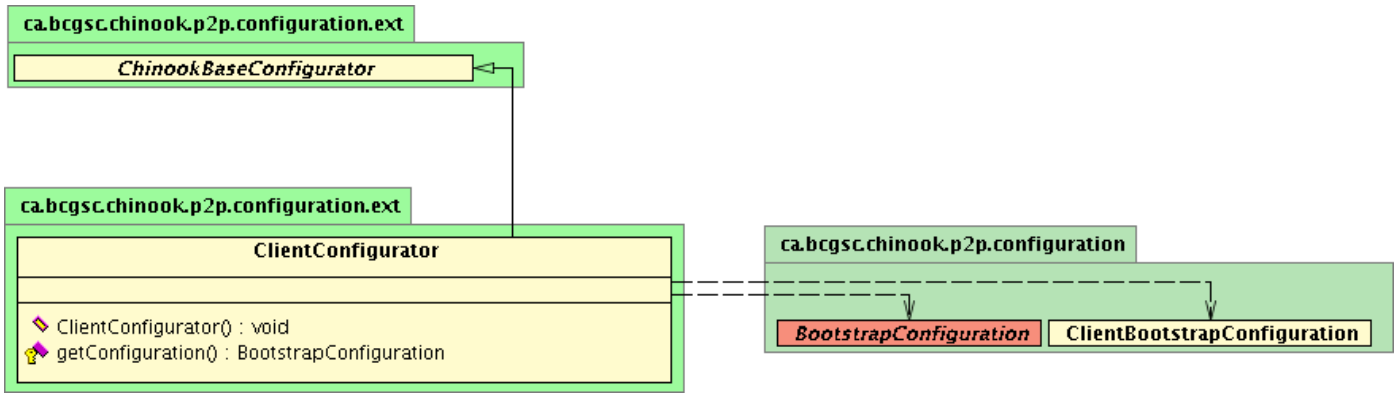

Figure 11: ClientConfigurator class

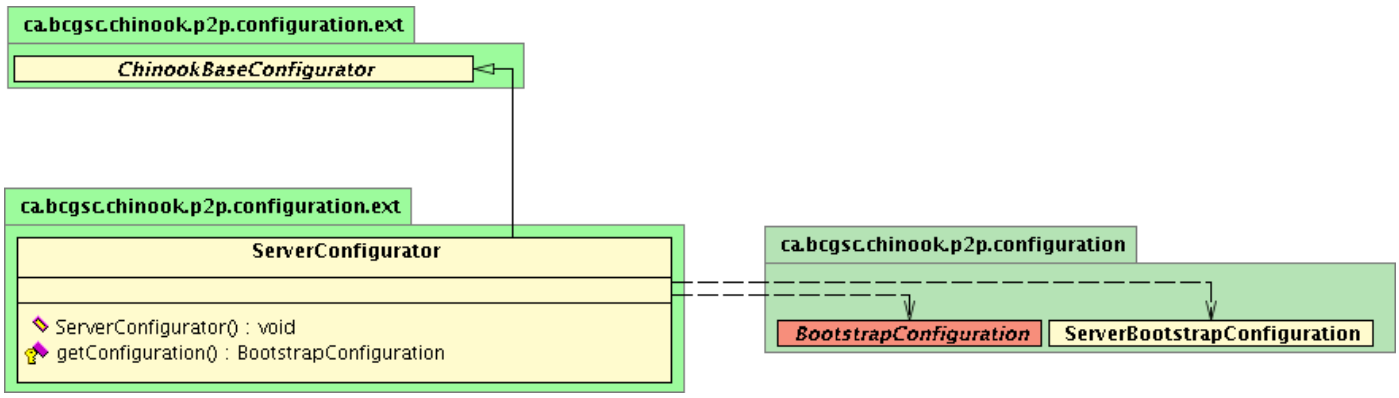

Figure 12: ServerConfigurator class