URQUi CAS Add On

1. Introduction.

- The URQUi CAS add on enhances the CAS single sign on software, by adding one time password (OTP) capability. OTP, greatly increases the security of the platforms CAS is being used to protect by eliminating the easily compromised static password operation.
- URQUi-CAS add-on is a new authentication handler that can be added to existing handlers or used on its own.
- The URQUi-CAS add-on is a free open-source software that can be modified to meet individual requirements. More information about URQUi, can be found at www.urqui.com.

2. Requirements specific for URQUi.

2.1. An existing functioning CAS installation.
2.2. The URQUi-CAS add-on had been tested to CAS release 3.5
2.3. Java policy files to enable unlimited strength encryption. These files and installation guidelines can be found at: http://www.oracle.com/technetwork/java/javase/downloads/jce-6-download-429243.html
2.4. A JDBC database (MySQL, Oracle, etc.) and associated API that can be used to store user information. This can be an existing database or create a new one if required.
2.5. A unique URQUi host identifier and associated key that can be found at: http://urqui.net/urqui_hostactivate.php.

3. Installation.

3.1. The URQUi-CAS add-on is available for download form GIT-Hub: https://github.com/urqui/cas. This contains the jar file as well as associated source.

3.2. Place the jar file in the {JAVA-SERVER-CAS} lib directory. This would be the same directory where the JDBC API exists.

3.3. Create a new table, or use an existing table, and have at least two columns, one that will contain the user identifier (such as screenname, email address), and another that will contain the users associated RQUi (the uses unique identifier to the URQUi system).

3.4. Retrieve the database access parameters, and user table credentials related to the table in 3 above. (dB name, table, login credentials). These need to be added (if not already present) to the deployerConfigContext.xml described below.

3.5. Modify the deployerConfigContext.xml to add the URQUi authentication handler. The handler can be added in sequence with other handlers or stand on its own. In Appendix 5.1 is an example of a standard implementation of the URQUi handler. Modify the attributes in {BLUE} to your own requirements.

4. Use.

4.1. The URQUi authentication handler uses the same credentials input as the standard CAS Client.

4.1.1. Username: This is the username that would be matched to the column in the validation table created in 3.3.

4.1.2. Password: This would be the URQUi PIN found on the main screen of the URQUi phone app on the user's mobile device.

4.2. The authentication handler, will try and retrieve the users RQUi from the validation table by matching the input username to a row in the table.

4.3. The authentication handler will then pass both the RQUi and URQUi PIN to the URQUi validation server for authentication. A TRUE will be returned for valid input, FALSE for invalid input.

5. Appendix.

5.1. deployerConfigContext.xml

```
<bean class="org.kissi.urqui.urquicas.URQUiAuthenticationHandler">

<property name="dataSource" ref="dataSource" />

<property name="sql"

        <value="select {rqui field name} from {user table} where lower( {user
        identifier} ) = lower(?)" />

 <property name="url">

        <value> {URL for Validation } </value>
```

```
</property>
 <property name="urquiid">
            <value> {URQUi Host Id } </value>
</property>
 <property name="urquikey">
            <value> {URQUi Host Id Key} </value>
</property>
</bean>
```