IBM Research - Zurich GmbH

Saeumerstrasse 4
8803 Rueschlikon
Switzerland

Author: Bernhard Jansen, Carl Binding
email: bja@zurich.ibm.com, cbd@zurich.ibm.com
Date: May 27, 2012
Version: 1.0

Subject: Input on the Real-Time Price distribution protocol for Ecogrid EU WP 3 Task 3.6.

1. Introduction

The document describes input to WP3.6 of the EcoGrid EU project. It describes the data, security and scalability requirements to a Real-Time Market Price and Real-Time Market Price Forecast application layer message format. Further on it compares a few transport and session layer protocols for broadcasting the price information to the user and presents a combination of protocols to achieve a highly scalable system to supply all customers of a TSO with a Real-Time Market Price.

2. Requirements

2.1. Data Requirements

To allow a working Real-Time Market system the Real-Time Market Price needs to be delivered with complete all attributes. The specified attributes are:

- The Real-Time Market Price
- The currency of the Real-Time Market Price
- The energy unit for the Real-Time Market Price
- The start and end time for which the Real-Time Market Price is valid (UTC)
- Forecast for the Real-Time Price consisting of any number of forecast values comprising a
    - Real-Time Price Forecast
    - The currency for the Real-Time Price Forecast
    - The energy unit for the Real-Time Price Forecast
    - The start and end time for a which the Real-Time Market Price Forecast is valid

2.2. Security Requirements

The distributed Real-Time Market Price is public information therefore it is not needed to encrypt the information. While there is no need to protect the information from being read by anybody other security requirements need to be assured. Namely

- *Authenticity* – The information has it origin with which says it is the source of the information
- *Integrity* – The information is not altered while being transported
- *Non-repudiation* – The source can't deny having said something
- *Non-replay able* – The information is only valid in one point in time. Information can't be recorded and replayed at different point in time.

2.3. Scalability Requirements

The system should scale from some thousands of Real-Time Market Price receivers (EcoGrid Consumers) to a very large number. The upper limit is the number of private households and participating businesses per TSO. This would be about 2.5[1] millions private households in Denmark under control of the Danish TSO Energinet plus the amount of participating businesses.

[1] http://www.statistikbanken.dk, as of 2004

2.4. Availability Requirements

The availability of the Real-Time Market Price signal is a crucial point to ensure a working Real-Time Market system and with a growing number of participants also for system stability. Outages in price distribution known to the Real-Time Market and the TSO are less critical as the TSO can act and take countermeasures with traditional regulation power. Outages in price distribution, for a significant amount of EcoGrid Consumers, which are unknown to the Real-Time Market and TSO are critical as the predicted reaction to Real-Time Market Price signal will significantly deviate from reality. In such a case the TSO will need to counter the situation by the use of traditional balancing power to ensure system stability. In the latter case the stress on the primary and secondary reserves is higher as they need to run longer. The TSO calculates with the reaction of the EcoGrid Consumers as predicted. If the reaction is smaller as many EcoGrid Consumers are not receiving the new Real-Time Market Price the TSO has to sense this and counter it with traditional reserves.

Unknown outages can be caused by:

- ISP outages of Ecogrid Consumers
- Firmware errors in standard router models of Ecogrid Consumers
- Denial of Service Attacks against distribution points

3. Format Description

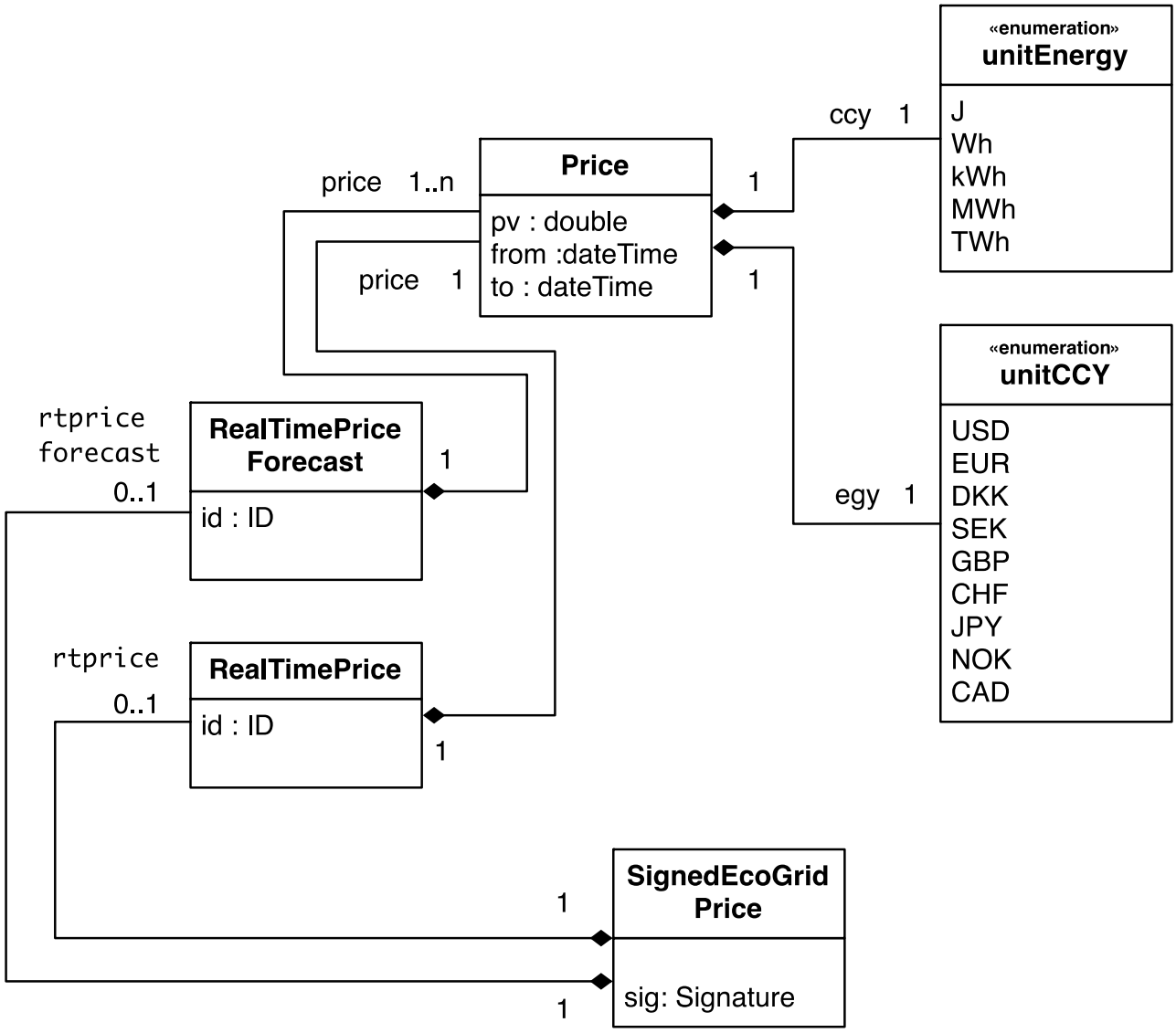
To allow an easy adaption of transport protocols the application level protocol is independent of the transport and session layer protocol. All data and security requirements are covered by the herein proposed application layer message format.

3.1. Message format and encoding

The required data, described in 2.1, is represented as a XML structure, which is described as an XML Schema. The XML Schema can be found in the Appendix. An UML representation follows in 3.2.

The format is generalized and allows the use within the EcoGrid EU project and for projects with a similar problem statement. It allows to send a price (exactly one) for energy, valid for a given time window, in a specific unit and in a given currency. The timestamps are given in UTC. A message can contain a price and or a price forecast. A price forecast contains a list of prices, each valid for a specific time window, in a specific unit and currency.

3.2. UML Class Diagram representation



The UML Class Diagram representation shows detailed view of the proposed message format description

3.3. Security

The security requirements stated in 2.2 are provided using a signature of the message in combination with a PKI infrastructure and the information of the message. As message signature schema the standardized W3C XML Signature Syntax and Processing is used. The signature and the message syntax provide the following security primitives.

- *Authenticity* is provided by the signature, as the signature contains the certificate of the signee which is signed by a trusted third party.
- *Integrity* is provided as the signed hash of the message can be recomputed, compared against the signed hash of the message and therefore it can be proven that it wasn't tampered after the signee signed the message.
- *Non-repudiation* is provided by the signature as the signee is not able to deny to have signed the message with the price information for a given time. The certificate with the signees' identity is attached to the signed message.

- *Non-replay able,* a message is not replay able as a valid message would be either declined by the client as being out of date proven by the start timestamp in the message or being declined as an attacker modified the timestamp and therefore breaks the integrity of the message.

A typical weakness of PKI infrastructures also applies here. The security primitives can only be guaranteed as long as the signee of the message and the Certificate Authorities private keys are not compromised. If the signee's private key is compromised the CA needs the revoke the certificate. The client therefore should be able to check an Online Certificate Revocation List or an Online Certificate Status Protocol Server. The use of cryptographic hardware to secure the private key is recommended.

4. Transport Protocol

The message format is deliberately independent of the transport and session layer protocol to allow the usage of the best fitting transport and session layer protocol for each situation.

4.1. Publish Subscribe Protocols

The problem statement to distribute a piece of information (here the Real –Time Market Price) from one source to a multiplicity of clients (here the EcoGrid Consumer) hints the usage of a Publish/Subscribe protocol. The advantages of such a protocol are that it is event based network load due to polling therefore is omitted, mostly provide guaranteed message delivery and give the possibility to know the number of subscribed parties. A disadvantage is the limited scalability, which can be countered by a tree structure to scale up to the necessary number of clients.

4.1.1. MQ (Message Queuing or Message Oriented Middleware)

Message Queuing or Message Oriented Middleware is not standardized as a protocol but provides standardized interfaces to different programming environments. Most prominently the Java Message Service (JMS) API. Such a message broker provides Publish Subscribe services with low overhead, rapid development time and provides additional features like guaranteed message delivery (as in TCP) and ways to add redundancy to message broker. As only the interfaces are standardized there is a variety of implementation in the Market. The Author is familiar with Apache Active MQ and IBM Websphere MQ. Both support JAVA, .NET, HTTP Clients and more.

4.1.2. XMPP PubSub

The Extensible Messaging and Presence Protocol (XMPP, RFC6120-6122), better known as the Jabber Internet chat protocol, has an extension to support Publish Subscribe. The extension is described in the Draft Standard XEP-0060 of the XMPP Standards Foundation. The protocol extension is quite new and the support for programming languages, in terms of libraries, is limited at the moment.

4.1.3. SIP Subscribe Notify

The Session Initiation Protocol, mostly known for Internet telephony, also supports Publish Subscribe but it is called Subscribe and Notify here. It is described in RFC 3856 and is originally designed to communicate on- and offline indicators. SIP, including Subscribe

and Notify is broadly used for VoIP and is a mature technology. As for Real-Time Market Price distribution the used feature set of the protocol would be very limited the overhead of using it would be very high. Although SIP is a open protocol fully described by RFC the amount of publicly available libraries to be used is limited.

4.2. IP Multicast

IP Multicast (RFC2236, RFC3170) is a very scalable way to broadcast data in an IP network. The scalability is reach by letting the network itself replicate the message to all receivers. The main drawback is the lacking support for it in the general Internet as for example the addressing is unclear. Most Multicast networks exist within networks of bigger ISP to provide high data rate services like IPTV in an economic way.

4.3. Combined technologies for scalability – Publish Subscribe and IP Multicast

One way to technically easy scale up to hundreds of thousands clients would be combine a Publish Subscribe technology and IP Multicasting and work, as a Real-Time Market Operator, jointly with the ISP. The Real-Time Market Operator would provide the Real-Time Price Signal to the ISP by using a Publish Subscribe technology and the ISP would push the Real-Time Market Price by using IP Multicasting within their networks to the customer. For ISPs with an existing IP Multicast infrastructure the effort is fairly minimal and the network load would stay fairly low.

Centralized control aggregators could either connect to the IP Multicast stream or via Publish Subscribe, depending on the number of clients and interface preference.

4.4. Proposed technology to be used in EcoGrid EU

The scalability requirements for the EcoGrid EU demonstration project much lower as described above for a complete TSO area. Overall 2000 or less EcoGrid Consumers need to be provided with a Real-Time Market Price. Therefore the use of a Publish Subscribe technology scales well enough is preferred.

As there exist many different Publish Subscribe technologies and within this category there are many different options to choose we propose the use of MQ as most the MQ Broker support interfaces to many languages.

5. Appendix

5.1. XML Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema xmlns="http://www.w3.org/2001/XMLSchema"
        xmlns:xsd="http://www.w3.org/2001/XMLSchema"
        xmlns:jxb="http://java.sun.com/xml/ns/jaxb"
        jxb:version="2.1"
        xmlns:sig="http://www.w3.org/2000/09/xmldsig#"
        targetNamespace="http://www.eu-ecogrid.net/pricing"
        xmlns:tns="http://www.eu-ecogrid.net/pricing"
        elementFormDefault="qualified">

        <!--  we have a local copy of this namespace's schema so we can import it. -->
        <import namespace=http://www.w3.org/2000/09/xmldsig#
                schemaLocation="./xmldsig-core-schema.xsd"/>

        <xsd:annotation>
                <xsd:appinfo>
                <jxb:schemaBindings>
                                <jxb:package name="com.ibm.zrl.ecogrid.jaxb.pricing"></jxb:package>
                        </jxb:schemaBindings>
                </xsd:appinfo>
        </xsd:annotation>

         <simpleType name="unitEnergy">
          <restriction base="string">
                 <enumeration value="J"/> <!-- [Ws] == [J] -->
                 <enumeration value="Wh"/>
                 <enumeration value="kWh"/>
                 <enumeration value="MWh"/>
                 <enumeration value="TWh"/>
          </restriction>
        </simpleType>

         <!-- 3 letter currency symbol ISO 4217 -->
         <simpleType name="unitCCY">
          <restriction base="string">
                 <maxLength value="3" />
                 <enumeration value="USD"/>
                 <enumeration value="EUR"/>
                 <enumeration value="DKK"/>
                 <enumeration value="SEK"/>
                 <enumeration value="GBP"/>
                 <enumeration value="CHF"/>
                 <enumeration value="JPY"/>
                 <enumeration value="NOK"/>
                 <enumeration value="AUD"/>
                 <enumeration value="CAD"/>
          </restriction>
        </simpleType>

         <!--  a singleton price which states the amount of energy, the currency, the price of the
                        energy-unit and the validity of the price.
          -->
         <complexType name="Price">
                <sequence>
                        <element name="pv" type="xsd:double"></element>
                </sequence>
        <attribute name="ccy" type="tns:unitCCY" use="required"></attribute>
        <attribute name="egy" type="tns:unitEnergy" use="required"></attribute>
                <attribute name="from" type="xsd:dateTime" use="required"></attribute>
        <attribute name="to" type="xsd:dateTime" use="required"></attribute>
    </complexType>

    <!--
        A set of one or multiple prices.
     -->
    <complexType name="RealTimePrice">
                <sequence>
```

```
                <element name="price" type="tns:Price" minOccurs="1" maxOccurs="1"></element>
            </sequence>
            <!-- mandatory ID for prices so that signature can reference the data to be signed-->
            <attribute name="id" type="xsd:ID" use="required"></attribute>
</complexType>

 <complexType name="RealTimePriceForecast">
            <sequence>
                <element name="price" type="tns:Price" minOccurs="1"
                maxOccurs="unbounded"></element>
            </sequence>
            <!-- mandatory ID for prices so that signature can reference the data to be signed-->
            <attribute name="id" type="xsd:ID" use="required"></attribute>
</complexType>

<!--
    the root element for signed prices.

    We propose a detached signature scheme. I.e. the <sig:Signature> element
    uses a <sig:Reference> onto the data being signed. Identification is via
    (relative) URI used as fragment identifier of the "id" attribute of the
    <Prices> element, e.g.
    <sig:Reference URI="#foo"></sig:Reference> references the
    <Prices id="foo">...</Prices> element.
-->
<element name="SignedEcoGridPrice">
            <complexType>
                <sequence>
                    <!--  we either send a singleton price ex-or a price forecast -->
                    <choice>
                        <element name="rtprice" type="tns:RealTimePrice" minOccurs="1"
                        maxOccurs="1"></element>
                        <element name="rtpriceforecast"
                type="tns:RealTimePriceForecast" minOccurs="1" maxOccurs="1"></element>
                    </choice>
                    <element name="signature" type="sig:SignatureType"></element>
                </sequence>
            </complexType>
    </element>









    <!-- the root element for unsigned prices -->
    <element name="EcoGridPrice">
            <complexType>
                <sequence>
                    <choice>
                        <element name="rtprice" type="tns:RealTimePrice" minOccurs="1"
                        maxOccurs="1" ></element>
                        <element name="rtpriceforecast"
                    type="tns:RealTimePriceForecast" minOccurs="1"
                    maxOccurs="1"></element>
                    </choice>
                </sequence>
            </complexType>
    </element>

</schema>
```