Demand Response Automation Server 3.x

Web Service Application Program Interface

Table of Contents

Preface

1.1 Purpose, Audience, and Scope

This document describes the Application Programmer Interface (API) to securely connect and listen to the DRAS Web Service (DRASWS) using SOAP to transport messages for the purposes of Automated Demand Response (AutoDR) using Open AutoDR (OpenADR) systems.

The API described in the document is supported by version 3.x of the DRAS.

1.2 Document Revision History

Doc. Version	Who	Date	Description
0.5	DH	5/19/05	Initial Revision
0.6	DH	6/11/05	Removed WSDL File Text
1.0	DH	7/29/05	Added URL Updated security information
2.0	DH/GG	5/25/07	Replaced LBNL header with simpler/shorted header Changed PSS2 to DRAS Changed prices to levels Updated the web service end points Added getPrice and isAPEEventPending calls Added DRAS version 3.5 revision information Added OpenADR implementation concept for AutoDR

1.3 Definitions, Acronyms and Abbreviations

- CPP - Critical Peak Pricing.
- PSS2 - Price Server System 2.0, name of the DRAS prior to 2006
- DB - Demand Bid.
- DRASWS -Demand Response Automation Server Web Service
- DRAS Client - The secure software on client's computer polling DRAS at the participant's site.
- EMS - Energy Management System
- AutoDR – Fully-Automated Demand Response using proprietary technologies and/or software systems.
- OpenADR - AutoDR implementation using an open standard, platform-independent, and transparent end-to-end technologies and/or software systems.
- Participant – Commercial electricity customer with a EMS requesting AutoDR signals from a DRAS Client

1.4 References

1. Pier Demand Response Research Center Web Site, Automated Facility Demand Response Page, http://drrc.lbl.gov/drrc-1.html
2. Simple Object Access Protocol (SOAP), version 1.1, http://www.w3.org/TR/soap/

3. Web Service Description Language (WSDL), version 1.1, http://www.w3.org/TR/wsdl
4. Extensible Markup Language (XML), http://www.w3.org/XML/

2 Introduction

The Demand Response Automation Server (DRAS) delivers DR signals and events-related service information electronically using the Service Oriented Architecture (SOA) to participants that have signed up for AutoDR programs within their respective utilities. Examples of current AutoDR programs are Automated Critical Peak Pricing Program (AutoCPP) and Automated Demand Bid Program (AutoDBP).

As shown in figure 1 in DRAS system architecture, the AutoDR participants will run the secure, platform-independent software (DRAS Client) on a client computer at their site that communicates with the DRAS securely over the Internet using XML-based web service technology. This software will also interface to the site's energy management system (EMS) to control electric loads based on participant's pre-defined DR strategies and program event information retrieved from the DRAS.

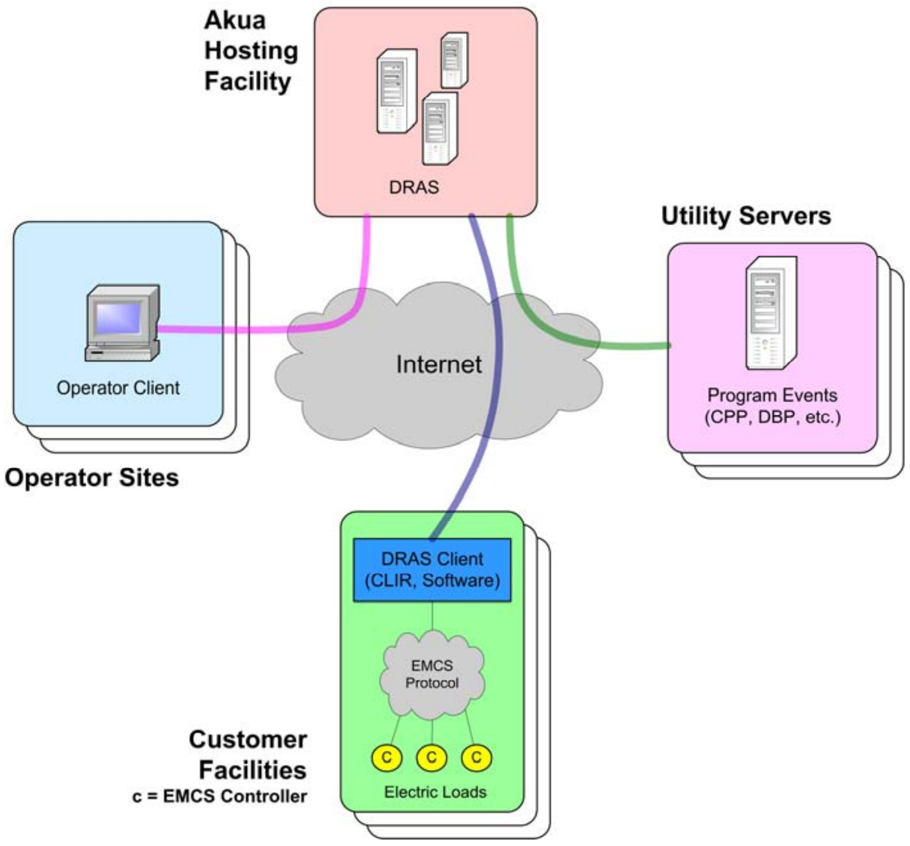

Figure 1: DRAS System Architecture

AutoDR Program events will be triggered by utilities based on parameters such as weather and total peak demand forecasts, which are specific to utility territories. AutoDR event information sent from the DRAS to the DRAS Client includes the current program level, the time, the value of any published changes in level that will occur in the future, and whether there are any program events pending (coming up soon). The program level can be one of three values depending on the details of the program tariff: normal, moderate, and high price-levels. These levels are typically used to control three corresponding shed levels in the EMS at the facility.

Note that in addition the electronic delivery of event information to the DRAS clients, in most cases either the utility or the DRAS will send an e-mail and/or e-page notification to the site's facility manager when an AutoDR event is issued (for both day-ahead and day-of notifications).

3 Protocol and Discovery

The DRASWS is a Web service with a SOAP API. DRASWS uses SOAP version 1.1 running on top of HTTP or HTTPS transport to request AutoDR service signals for current programs.

The DRASWS does not support any discovery mechanism. Participants will be supplied with a username and password, and a Web Services Description Language (WSDL) file that defines the Web service based upon the 1.1 version of the WSDL standard.

4 Web Service End Point

The DRASWS end point is https://*servername*/PSS2WS/PSS2WS

The *servername* depends on the electrical utility servicing the participant's facility as follows:

- Pacific Gas & Electric (PG&E) - pge.openadr.com
- Southern California Edison (SCE) - sce.openadr.com
- San Diego Gas & Electric (SDG&E) - sdge.openadr.com

5 SOAP Messages

The DRASWS supports three request/response service messages using SOAP:

- *getPrice* - retrieves the current program level
- *getPriceSchedule* - retrieves the current program level as well as a list of changes to the level that will occurring the future.
- *isAPEEventPending* - indicates if a program event will start be starting within the next 21 hours.

The *getPrice* and *getPriceSchedule* requests include the last value retrieved from the DRAS as an aid in tracking communication errors (any errors are reported to the DRAS operators).

The details of this message are described in the following sections.

5.1 Levels versus Prices

Although the DRAS API refers to prices, these values are actually a fixed set of real numbers (double) that correspond to levels:

- 1.0 - normal level (default)
- 3.0 - moderate level
- 5.0 - high level

The price field in all messages will always have one of these three values (1.0, 3.0, or 5.0).

5.2 Types

The following types are used in the DRASWS SOAP messages for services.

5.2.1 PriceScheduleEntry

The PriceScheduleEntry type represents a change in price at a given time.

Field	**Type**	**Description**
date	dateTime	Time when the new level becomes active. UTC.
priceDPKWH	double	New program level: • 1.0 - normal level • 3.0 - moderate level • 5.0 - high level

5.2.2 PriceSchedule

The PriceSchedule type represents the current price as well as the list of price changes that will occur in the future.

Field	**Type**	**Description**
currentPriceDPKWH	double	Current level: • 1.0 - normal level • 3.0 - moderate level • 5.0 - high level
entries	Array of PriceScheduleEntry	Array of PriceScheduleEntry types, each one representing a level change. Sorted by time in ascending order.

5.3 Messages

5.3.1 getPrice

Request the current PriceSchedule.

Field	**SOAP Type**	**Description**
lastPrice	double	The level returned from the server in the last call to getPrice.

The response to getPrice contains the current level.

Field	**SOAP Type**	**Description**
price	double	Current level: • 1.0 - normal level • 3.0 - moderate level • 5.0 - high level

5.3.2 getPriceSchedule

Request the current PriceSchedule.

Field	SOAP Type	Description
lastPriceSchedule	PriceSchedule	The PriceSchedule returned from the server in the last call to getPriceSchedule.

The response to getPriceSchedule contains the current price schedule.

Field	SOAP Type	Description
priceSchedule	PriceSchedule	Current PriceSchedule.

5.3.3 isAPEEventPending

Request the current isAPEEventPending. The request has no arguments

The response to isAPEEventPending indicates if a program event will start within the next 21 hours.

Field	SOAP Type	Description
eventPending	boolean	*true* if a program event will start within the next 21 hours

6 Usage

Participant clients should poll getPrice or getPriceSchedule on the DRASWS every 60 seconds. To ensure successful delivery of pricing information, there must be a minimum of 50 seconds and a maximum of 300 seconds between polls.

If needed by the EMCS control strategy, isAPEEventPending should also be polled along with either getPrice or getPriceSchedule.

7 Security

Access to the DRASWS will be done using Secure Socket Layer (SSL) communications that provides server authentication, data encryption and data integrity. Client authentication is done using HTTP authentication. The utility will provide each participant with a self-signed digital certificate, a username, and a password that will be used for all the communications with the DRASWS.

8 Contact

Lawrence Berkeley National Laboratory; Demand Response Research Center (DRRC): DRRC@lbl.gov

Akuacom Inc: info@akuacom.com