Demand Response Analysis and Control System

Reference Architecture and Design

Prepared For:
California Institute for Energy and Environment
University of California Office of the President

In Response To:
Research Opportunity Notice DR ETD-02-01

Authored By:

EnerNex Corporation
620 Mabry Hood Road, Suite 300
Knoxville, TN 37932
Phone: +1-865-218-4600
Fax: +1-865-218-8999
E-mail: info@enernex.com

www.enernex.com

November, 2008

Document History

Revision History

Revision Number	Revision Date	Revision / Reviewed By	Summary of Changes	Changes marked
0.1	01/20/09	Stuart McCafferty	Original Document	N
0.15	2/17/09	Stuart McCafferty	Revisions	N
0.16	2/19/20	Jerry Melcher	Final Draft Revisions	Y
0.17	2/20/09	Stuart McCafferty	Final Draft for Internal Review	N
0.19	3/3/09	Jerry Melcher	Final Draft following Internal Review	Y
1.0	3/6/09	Stuart McCafferty	Final Document	N
1.1	3/12/09	McCafferty	Final – added planning module to DRMS figure	N

Approvals

This document requires the following approvals.

Name	Title
Gaymond Yee	CIEE Program Manager
Erich Gunther	EnerNex Chief Engineer

Contents

1. Introduction and Background 6
  1.1 Document Context 6
  1.2 Background 7
  1.3 Reference Architecture and Design 8
2. DRACS System Overview 9
  2.1 DRMS Parent System 9
    2.1.1 DRMS Information and Decision Flow 11
  2.2 DRACS Module 13
    2.2.1 Situational Understanding 13
    2.2.2 DRACS High Level Functionality 14
    2.2.3 DRACS Components 15
    2.2.4 DRACS Information and Decision Flow 16
3. DRACS Design Considerations 22
  3.1 Assumptions 22
  3.2 Dependencies 23
  3.3 General Constraints 24
  3.4 Goals and Guidelines 24
  3.5 Development Methods 25
4. Architectural Strategies 28
  4.1 Service Oriented Architecture 28
  4.2 Gridwise Architecture Council Interoperability Framework 30
5. DRACS Reference Architecture and Design 32
  5.1 Database 33

5.2 Demand Resource Situational Understanding Engine.......35
5.3 Visualization.......39
5.4 Demand Resource Attributes and Behavior Collection.......43
6. Policies and Tactics.......46
6.1 Enterprise Service Bus.......46
6.2 GridWise Architecture Council.......47
7. Glossary.......50
8. Bibliography.......52

Table of Figures

Figure 2-1: DRMS System – Information and Decision Flow....10
Figure 2-2: DRMS-DRACS Conceptual Relationship....12
Figure 2-3: DRACS Conceptual Components....15
Figure 2-4: DRACS Architecture - Information and Decision Flow....17
Figure 2-5: Participant Demand Response Services....18
Figure 2-6: Three Types of Baseline Models....20
Figure 3-1: DRACS dependencies....23
Figure 3-2: DRACS and DRMS instances will operate within CAISO, IOUs, Co-Ops, and DR Providers....25
Figure 4-1: SOA Provides Business Process Flexibility by Abstracting Functionality to Services....28
Figure 4-2: DRACS Minimal SOA Layers and Artifacts....29
Figure 5-1: DRACS Data Flow....32
Figure 5-2: DRACS database has 5 primary entities....34
Figure 5-3: DRACS Demand Resource Situational Understanding Engine component highlighted....36
Figure 5-4: Operational Demand Resource Situational Understanding Engine Subcomponent Block Diagram....37
Figure 5-5: DRACS computes confidence intervals using statistical methodologies in evaluating scenarios....38
Figure 5-6: Oak Ridge National Laboratory grid visualization using Google Earth....40
Figure 5-7: DRACS shows where the energy is, where it is being used, and where it is needed....41
Figure 5-8: Google Earth provides switchable layers similar to DRACS visualization operator options....43
Figure 6-1: ESB provides tactical solution for SOA strategy....46

1. Introduction and Background

1.1 Document Context

This document meets the deliverable requirement for Task 3, Develop Reference Architecture for a Demand Response Analysis and Control System (DRACS) for the California Institute for Energy and Environment's (CIEE) grant for Research Opportunity DR ETD-02-01. The grant was provided to the EnerNex team to address two research topics and questions identified in the opportunity notice:

- *"Using a military (or another, such as air traffic control) C3I (Command, Control, and Communication) system as a model, adapt it to conceptually deal with C2I electricity applications such as dispatching DER to keep the lights on. Compare and contrast the chosen C3I model with the requirements for implementing a C2I strategy for integrating utility information and communications systems. Are there analogies that indicate utilities can operate in a similar fashion? If not, what are the gaps that need to be filled and is it feasible to fill the gaps?*
- *Given current utility systems and assuming the systems are integrated, how would the CAISO or an UDC operate its control centers in a military (or another) C3I style? With up-to-date real-time information and the ability to control all of its available assets, given a particular operational scenario, how would a plan to address the scenario be executed using strategies based on military (or another) C3I?"*

C3I Approach

C3I fundamentally cuts across all operational elements of an organization and must function as a "single system"(different from most systems which partition more along physical lines). Historically, communications, networks, command and control, security, and information systems were designed and developed separately. Legacy systems optimized for given organization/mission vs. C3I systems which must accommodate multiple elements/platforms and be flexible to exploration of new styles of operation.

Layered approach
•Isolates change impacts (enabling evolution)
•Based on industry standards.
•Includes publish & subscribe messaging framework (enabling plug-n-play applications by establishing well defined data interfaces).

Interoperability
•Focus on standards and approaches that enable interoperability between elements.
•Establish small set of interface standards & reduce possible number of interface combinations.

•Requires interoperability at all layers: communications, networks, security, command, control, and information.

The reference architecture identified in this document is based on careful studies and (EnerNex Corporation, 2009), interaction with San Diego Gas & Electric stakeholders, and coordinated efforts between EnerNex Demand Response experts, the Open Smart Grid AMI Enterprise team, and C3I system engineers at Oak Ridge National Laboratories (ORNL).

1.2 Background

California is poised to implement a widespread demand responsive energy grid that includes a variety of technologies and tariffs as well as regulatory and social influences. If widely deployed, such a system would closely couple power system, communication, and "societal behavior" networks in such a way that the nonlinearities embedded in these systems may result in unforeseen interactions and chaotic behavior. This is illustrated by the August 10, 1996 West Coast blackout that resulted from a system disturbance that caused uncontrolled load shedding at PacifiCorp and subsequent instantaneous power swings of 6 GW throughout the western state's power grid. A proper C3I system would have prevented this system behavior

There exists an urgent need to understand and manage the overall complex system behavior of a demand responsive energy grid in order to provide real time (and faster than real time) monitoring, analysis, and control. This "understanding" must support aggregate behavior anticipation and analysis in terms of location of loads and utility assets, electrical grid and communication network topology, as well as cost structures and tariffs, information flow across the communications network and energy grid, and the resulting coupled control linkages.

The primary contributor to improved system performance is the ability to accurately predict in real-time the total amount of equivalent dispatchable resource available through DR – in essence a comprehensive Demand Response Analysis and Control System (DRACS). In this context, DR includes load management due to:

- pricing signals
- direct load control
- interruptible loads
- and distributed resources.
- control system modeling

Improved performance results by understanding the reliability and stability of the power and communication systems at all times, and then using that information to achieve specific performance goals. DRACS is the proposed solution for providing this situational understanding and analytical capability and is one of the primary software modules within an integrated Demand Response Management System (DRMS).

1.3 Reference Architecture and Design

This document is modeled after the Organization for the Advancement of Structured Information Standards (OASIS) SOA reference architecture guidelines. (Organization for the Advancement of Structured Information Standards (OASIS), 2008).

This Reference Architecture and Design document is an abstract representation of the DRACS system, focusing on the elements and their relationships needed to enable DRACS to be created and used within a demand response environment. The reference architecture avoids reliance on specific technologies, whenever possible, to allow vendors choices in their implementation (language, operating system, etc.).

DRACS is a system that requires “future-proofing” and will likely be used across IT system boundaries for extended periods of time in a Service Oriented Architecture (SOA) relationship. Vendors need to address not only how the system is to be constructed, but also how it will integrate with the Demand Response Management System (DRMS) and other systems. With the fluid nature of the demand response landscape, *flexibility, extensibility,* and *reliability* are important considerations when developing the reference architecture and design of DRACS.

2. DRACS System Overview

2.1 DRMS Parent System

The DRACS system can be viewed as a sub-system or module within the Demand Response Management System (DRMS). It is responsible for real time situational understanding and the current DR network topology.

The DRMS is out of scope for this document. However, there are some basic DRMS architectural and functional assumptions that are necessary for the DRACS reference architecture to make sense.

The DRMS is the overall management system for the demand response environment. In addition to the situational understanding functionality supported by DRACS, this includes making business decisions based on business rules on how, who, what, when, and where demand events occur. These business rules can be used for both real time DR events or in planning for day-ahead events. Based on these business rules, the DRMS is responsible for developing the optimal scenarios/combinations of demand resources for addressing each demand response event. Additionally, the DRMS communicates with these demand resources through systems such as AMI, legacy load control systems, and contracted DR providers.

Figure 2-1, below, is a simplified conceptual drawing of the overall DRMS system. The drawing is not meant to imply completeness or any concrete design. In fact, some obvious omissions in this drawing include a reporting functionality and a customer portal capability. Rather, the figure is provided primarily to show the critical functionality, and decision and communication flow that is anticipated for a complete DRMS system.

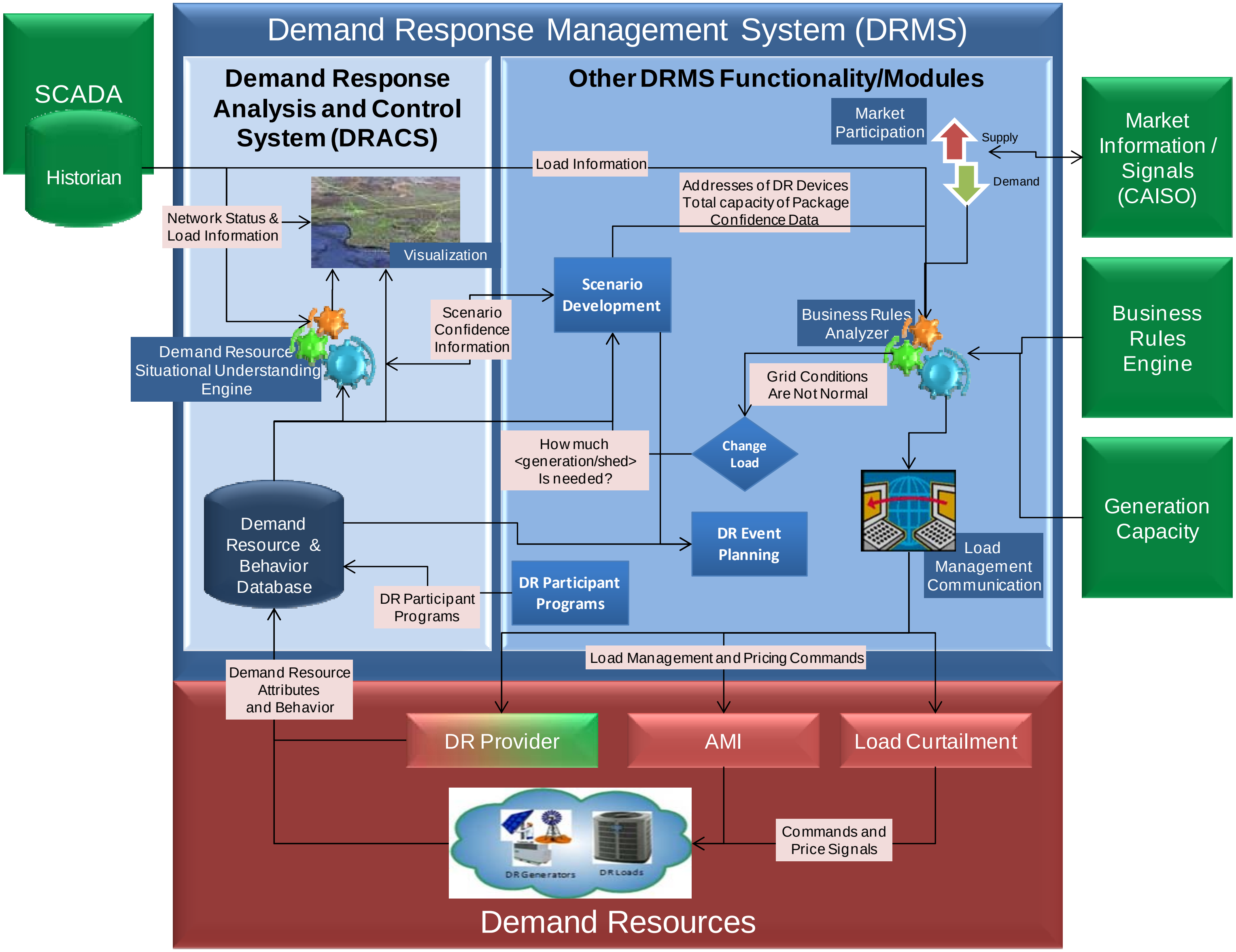


Figure 2-1: DRMS System – Information and Decision Flow

The green boxes in Figure 2-1 show external systems and messages which are inputs to DRMS. They include custom business rules for the organization, market and demand response commands from the RTO, in this case the CAISO (or the authoritative DR entity for the DRMS instance – for example, in an instance where an aggregator has his own DRMS, that authority may be a utility), and the current generation utilization and capacity within the system.

The red boxes depict output target systems or communication paths. They include AMI, existing load curtailment systems, and DR aggregators. Note that the DR Provider box is a blended green and red because it provides both input and output to the DRMS system.

The large blue box and the enclosed smaller blue boxes depict the DRMS system and include the DRACS module shown on the left. The DRACS module will be expanded and discussed in more detail in the following section. The Business Rules Optimization and Analysis box is where business rules are applied and commands are sent out.

2.1.1 DRMS Information and Decision Flow

The DRMS receives a DR event signal from the RTO or it independently determines that grid conditions are out of balance or may become unbalanced. The imbalance can be based on reliability conditions (conditions leading to potential electricity supply issues) or economic conditions (conditions where business rules determine there are non-optimal electricity cost/pricing proportions). The business rules drive all the decision-making capabilities of the DRMS. DRMS next determines the amount of load shed or generation needed to correct the imbalance. DRMS reviews the database of Demand Resources and determines a mix of resources that match the business rules, creating a “scenario” of demand resources. This scenario, a collection of demand resources, is passed to the DRACS module to determine the scenario appropriateness and probability of meeting the DR event requirements. DRACS returns this information and the DRMS can either execute the scenario or create another scenario for DRACS evaluation. This loop may occur several times before determining the optimal scenario for the DR event. Once the scenario has been selected, the DRMS executes the commands to communicate with the affected Demand Resources and/or contracted Demand Response providers. In the future, this communication is expected to primarily occur through the AMI system. However, legacy load management systems exist and will likely continue for years to come. The DRMS will need to understand the mechanism for communicating to each end point Demand Resource through the legacy and AMI systems. Once the DR event has been initiated, the DRMS monitors the grid conditions to determine whether additional adjustments are necessary. If an adjustment is necessary, the entire process described starts again.

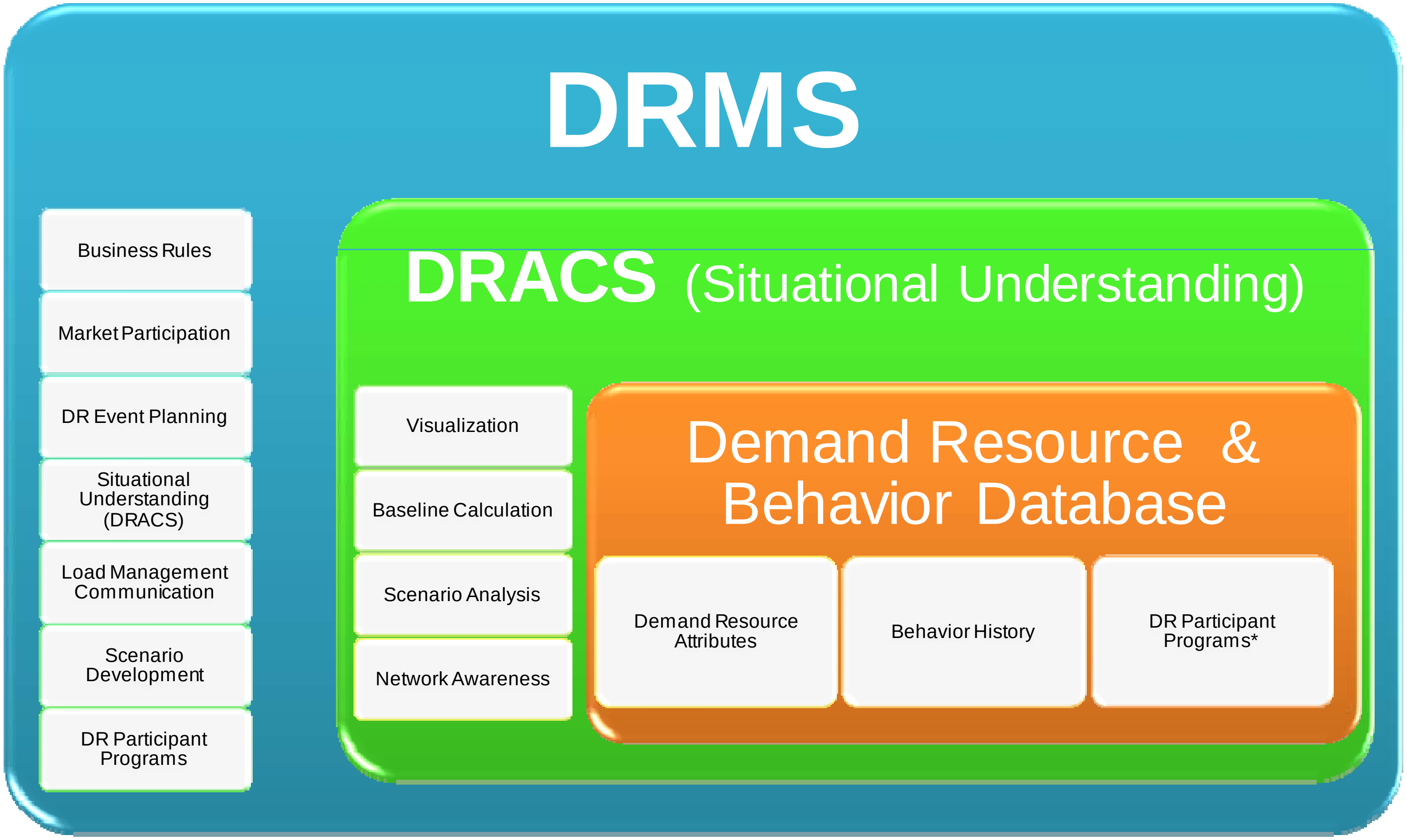


Figure 2-2: DRMS-DRACS Conceptual Relationship

Figure 2-2, above, shows the conceptual relationship of DRACS to the DRMS system. DRACS is one of the 6 modules within DRMS. The 6 modules include:

1. <u>Business Rules Analyzer</u>. This module uses business rules to determine when DR events should be initiated.
2. <u>Market Participation</u>. This module performs energy market analysis and management.

3. Demand Response Event Planning. This module performs day-ahead planning used to pre-select and optimize DR scenarios for known events.
4. Situational Understanding (DRACS). DRACS provides the real time demand resource and DR topology situational understanding.
5. Load Management Communication. This module provides the communication link to the various load management systems.
6. Scenario Development. This module uses business rules and DRACS simulator information to develop DR event scenarios.
7. DR Participant Programs. This module supports the management of demand response programs within the organization.

2.2 DRACS Module

The previous section described the DRMS system in total. This section provides greater detail DRACS features and functionality and how the DRACS module operates within the overall DRMS.

2.2.1 Situational Understanding

The DRACS module is responsible for the *situational understanding* of the demand response environment. This real time comprehension goes beyond just “awareness”. In other words, DRACS must support a real time assessment capability that provides detailed and accurate information on the entire demand responsive network topology. In addition, past behavior and historic patterns are part of the “understanding” necessary to calculate current capabilities and system response expectations. DRACS situational understanding includes:

- Current electrical network activity. This real time network activity includes awareness and understanding of weather, outages, voltage loss, congestion, or other pertinent information which can affect the ability to support a DR request. In addition, this information can be used to help predict the effectiveness of a DR request against a subset of DR customers (Demand Resources) or help to target the DR request to customers/locations capable of supporting the DR request more effectively.
- Existing demand response topology and historical behavior. Each active Demand Resource, its potential DR capacity (Baseline), and historical behavior patterns must be understood in order to develop effective DR request scenarios and implementation strategies. Demand Resources may be load managing, supply managing, or both. Customer behavior, such as percentage of time they “opt out” of DR events and amount of DR capacity supported in DR events, are important variables in predicting future behavior and in calculating a Baseline and the total expected response from each Demand Resource (or customer).

Increased visibility of performance data, ongoing monitoring of DR systems and resources, and the consistency of customer responses will be examined for their contribution to the effectiveness of the DRMS system through DRACS.

2.2.2 DRACS High Level Functionality

In order to accomplish the situational understanding requirements, DRACS supports the following functionality:

1. Observe (and visualize) real time network information network-related events (outages, voltage loss, etc.). This real time functionality is necessary in order to provide visual feedback of current relevant events and the overall real time DR topology.
2. Load and maintain database of demand responsive resources. A comprehensive, accurate list of Demand Resources must be maintained in order to provide the DR event predictive capabilities necessary in the DRMS system. The database must be scalable and built to support 100's of thousands or even millions of resources. In addition, DRACS will provide a visualization tool for the Demand Resource topology. The visualization component will provide the operator the ability to drill up and down to different levels within the DR topology, providing functionality similar to "Google Map".
3. Load and maintain database of demand responsive behavioral information. Each Demand Resource (or customer) historical response to DR requests must be captured for use in predicting future behavior. This information will be stored in a relational database and will be updated frequently to ensure optimal predictive information.
4. Receive and analyze scenarios for compliance with DR requests. DRACS will receive "scenarios" from other modules within the DRMS system, which have been created based on the organization's business rules. These scenarios will provide a desired load response, the timeline involved, and a set of DR customers and resources. These scenarios will be evaluated by the DRACS system to determine the likelihood (or confidence interval) of success in meeting the DR load response objective. DRACS will achieve this capability by employing standard network and historical pattern prediction algorithms and through analysis of customer behavior in responding to previous DR events.
5. Model the system – DRACS will provide visualization capabilities that overlays the Demand Response topology and interaction with the network topology. Preferably, DRACS will utilize existing topology models rather than requiring the creation of its own models. The models will be utilized by the real time operational system and an off line simulator.

2.2.3 DRACS Components

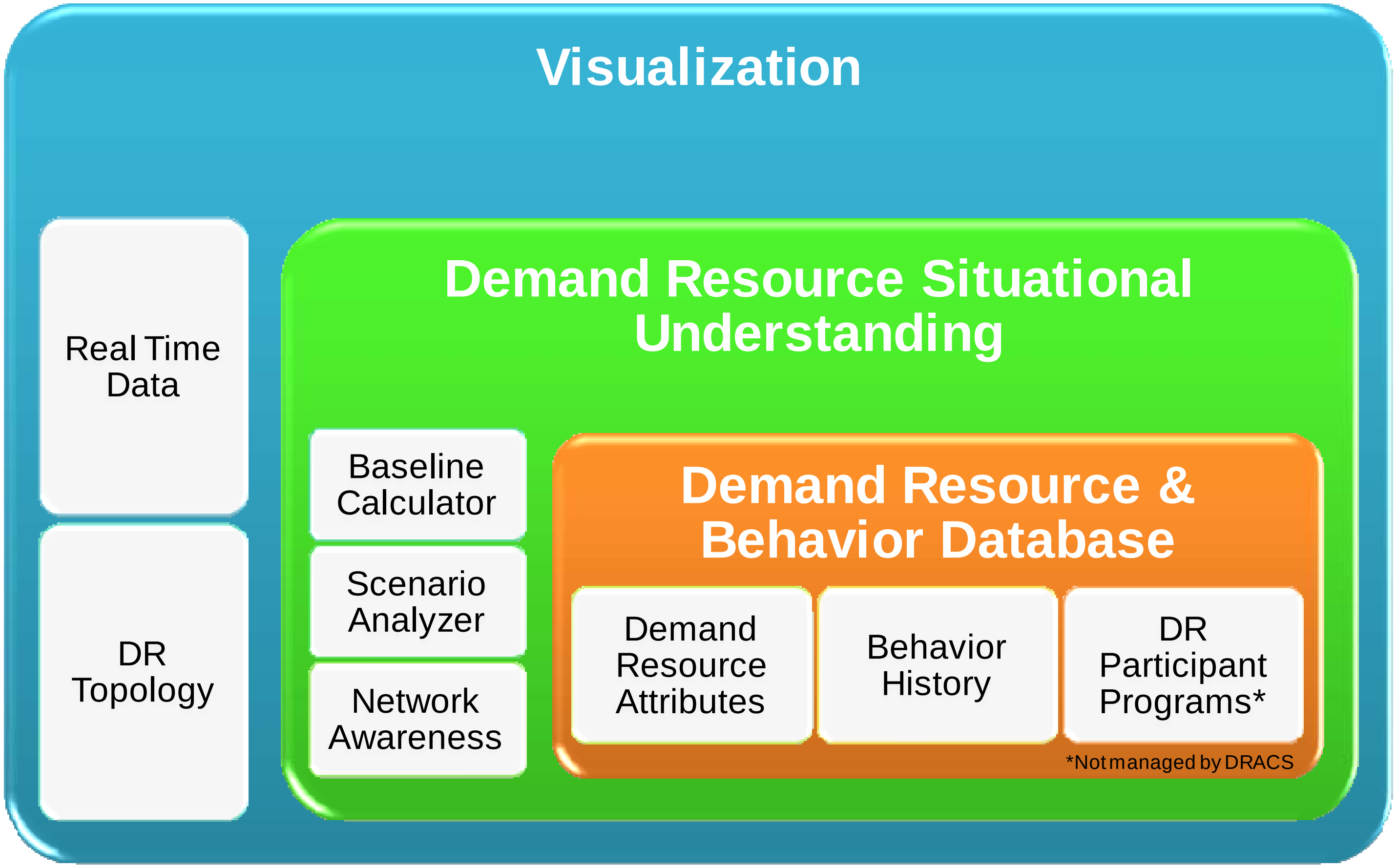


Figure 2-3: DRACS Conceptual Components

There are three (3) primary components within the conceptual DRACS architecture depicted in Figure 2-3 above:

1. Demand Resources and behavior database – this relational database houses Demand Resource information such as the device type, service type, customer ID, path/topology information, enrolled customer programs, maximum load, normal load, GPS location, and any other attributes pertinent to understanding the location and load profile of the resource. In addition, the database also houses behavioral patterns of the resource based on historical data. Also, DR event data is stored to understand the system's expected response vs. the actual response for each DR event. The existence and required

maintenance of this database implies an import capability that periodically captures behavior, and provides a mechanism for adding and removing Demand Resources.

2. Demand Resource Situational Understanding – this analytical tool is the heart of the DRACS system. It receives a scenario from the DRMS system, analyzes the scenario requirements against the Baselines and current network environment, historical behavior, and real time conditions, and then returns the probability of success to the DRMS system.
3. Visualization – the visualization component provides the operator with a real time topological view of the demand response landscape. It also provides the operator with a view of real time network events and a drill up/down capability for any Demand Resource that exists in the DR topology.

2.2.4 DRACS Information and Decision Flow

Figure 2-4, below, depicts a simplified DRACS architecture provided to show the critical functionality, and decision and communication flow that occurs within DRACS and complements the drawing and discussion from Section 2.1, DRMS Parent System. There are 3 different information and decision flows within DRACS. They include:

1. Network Status and Load Information
2. Demand Resource Attribute and Behavior Information
3. Scenario Confidence Information

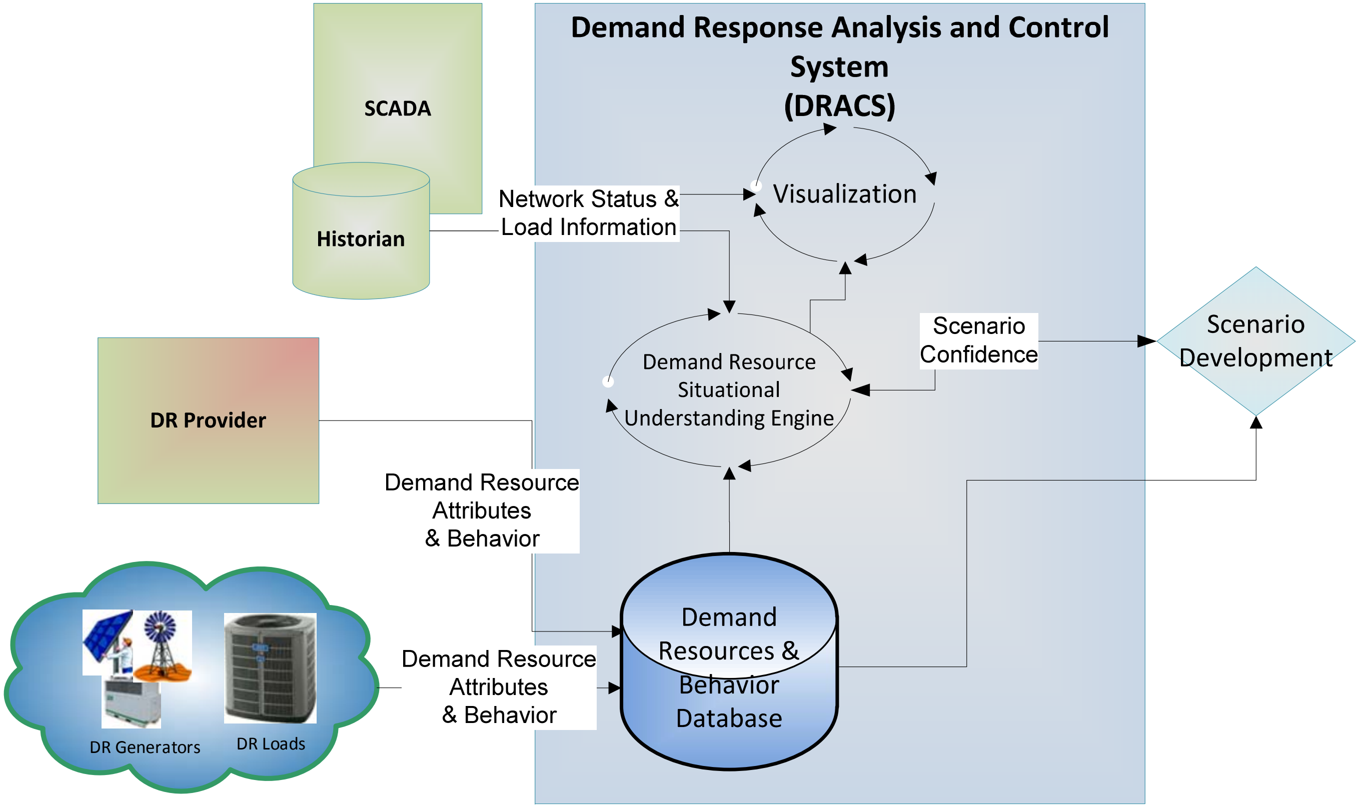


Figure 2-4: DRACS Architecture - Information and Decision Flow

2.2.4.1 Network Status and Load Information

DRACS constantly monitors the status and health of the electrical network through the SCADA historian. This information is superimposed on the Demand Resource network topology in order to understand the effect of network events (outages, communication losses, voltage drops, etc.) that may affect the ability of DRMS to respond effectively to DR events.

The DRACS visualization component provides a real time mechanism for viewing network status and health. It provides a “Google Map”-like visualization capability that allows the operator to drill up or down along the DR topology. It is not expected, nor is it wanted, for DRACS to provide its own topology design/development environment. The expectation is to utilize an existing network topology map (import from tools like Siemens Power System Simulator for Engineering (PSS/E)) to remove duplication of effort and minimize operator learning curve.

The Network Status and Load Information flow is one-way. The DRACS system receives network loads and health data. The data is viewable by an operator in real time using the visualization component. Network information is used by the Demand Resource Situational Understanding Engine as part of the algorithm for determining the probability of success for a scenario against a DR event.

2.2.4.2 Demand Resource Attributes and Behavior Information

DRACS is responsible for maintaining a relational database of demand resources, their attributes, and their historical behavior/patterns against DR events. Although there are several research and standards activities around Demand Response, there are currently no common mechanisms for auto-discovery of Demand Resources. Therefore, adding, modifying, replacing, and deleting Demand Resource information from the database will either be a manual process or a custom/proprietary solution and will be developed on a case-by-case basis for each DR system or vendor solution. As these research discoveries and standards mature, automated mechanisms will be retro-fitted to the DRACS system.

Customers will agree and sign-up to participate in one or more Demand Response programs. Utilities and Demand Response Providers will typically interface with customers using a Customer Relationship Management System (CRM). The CRM will provide information to the appropriate Demand Response Program. The DR Program participants will have Demand Resource equipment and services installed at their location. In support of these DR Programs, participants will provide one or more of the following services:

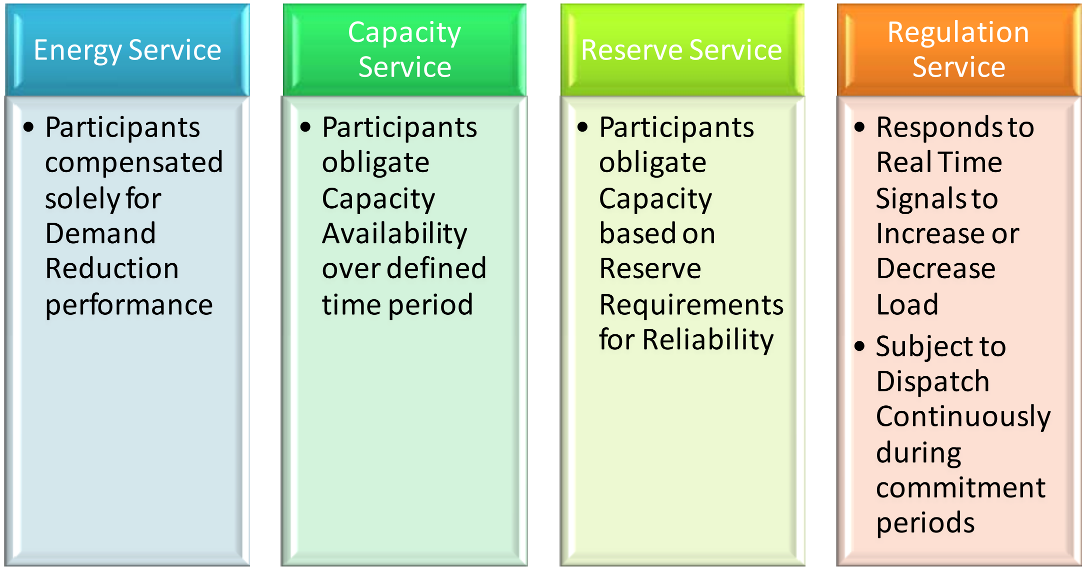


Figure 2-5: Participant Demand Response Services

1. Energy Service - A type of Demand Response service in which Demand Resources are compensated based solely on Demand reduction performance. This is the most common type of Demand Response service.
2. Capacity Service - A type of Demand Response service in which Demand Resources are obligated over a defined period of time to be available to provide Demand Response upon deployment by the System Operator.
3. Reserve Service - A type of Demand Response service in which Demand Resources are obligated to be available to provide Demand reduction upon deployment by the System Operator, based on reserve capacity requirements that are established to meet applicable reliability standards.
4. Regulation Service - A type of Demand Response service in which a Demand Resource increases and decreases Load in response to real-time signals from the System Operator. Demand Resources providing Regulation Service are subject to dispatch continuously during a commitment period. Provision of Regulation Service does not correlate to Demand Response Event timelines, deadlines and durations.

Until automated discovery services for Demand Resources mature, field service technicians provide support in linking the participant Demand Resources and services to the DR program. Customer enrollment business processes should be in place to add the participant, Demand Resources, and enrolled DR programs to the DRACS database. Deletion and replacement of Demand Resources and participants from the DRACS database is also a manual process with established business processes and may be handled by customer service or made available through a web portal. The manual nature of adding, deleting, and replacing Demand Resources and participants is a real concern as the Demand Resource numbers increase. The more accurate the attribute and behavior information in the database, the more accurate DRMS will be in managing DR events.

Updating Demand Resource behavior is an asynchronous activity that collects participant and Demand Resource reactions against DR event participation requests. Information such as opt outs and load reductions/increases is collected in order to provide historical data to help predict future responses and anticipated levels of event support. For legacy load control systems, this data will be collected through existing business processes and systems. For DR systems using AMI meters and communication routes, DRACS will acquire post-event behavior data from Meter Data Management (MDM) systems or from the Customer Information System (CIS). There is no requirement to collect this information in real time. Behavioral information will provide metering information on load changes that occurred during DR events. This may be the difference between pre-event loads and loads concurrent with the event or metered values pre-event and concurrent. Post-event response behavior is necessary for billing and scenario prediction purposes only, and is not necessary for calculating real time load management expectations from the Demand Resources participating in a DR event. In other words, a daily or billing cycle update of behavior information is adequate for the needs of the DRACS system.

The information flow for both Demand Resource attributes and behavior information are one-way communication, originating from the Demand Resource and finishing at the Demand Resources and Behavior Database in DRACS. DRACS designers/developers must ensure that a tool/utility, web services, and set of processes exists for adding, deleting, replacing, and modifying Demand Resource attributes and that these functions are included as part of the overarching DRACS solution. A similar set of services is necessary for Demand Resource behavior. These utilities may be included within the DRACS core or as separate applications since they are expected to change to support emerging Demand Response standards, communication mechanisms, and Demand Resource devices.

2.2.4.3 Scenario Confidence Information

The DRMS generates scenarios for managing each DR event. These scenarios are built based on business rules and from a collection of Demand Resources in the Demand Resource Attribute and Behavior database that meets both the business rules and DR event requirements.

Scenario confidence information flow is bi-directional communication with the DRACS Demand Resource Situational Understanding Engine as the central Actor in the flow. It receives a scenario from other modules within DRMS. It then evaluates the scenario against the collection of Demand Resources Baseline estimates and previous behavior, current electrical network events, and topology communication and social response algorithms and calculates the likelihood of success in meeting the DR event objectives.

A Baseline is an estimate of the electricity that would have been consumed by a Demand Resource in the absence of a Demand Response Event. Baseline is compared to the actual metered electricity consumption during the Demand Response Event to determine the Demand Reduction Value. Depending on the type of Demand Response product or service, Baseline calculations may be performed in real-time or after-the-fact.

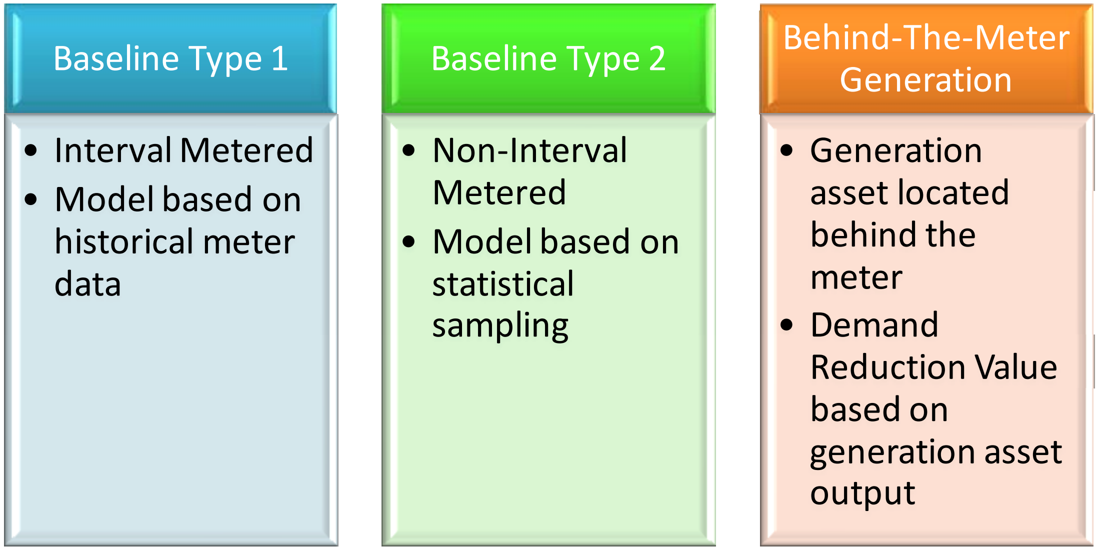


Figure 2-6: Three Types of Baseline Models

1. Baseline Type 1 (Interval Metered) - A Baseline model based on a Demand Resource's historical interval meter data which may also include but is not limited to other variables such as weather and calendar data.
2. Baseline Type 2 (Non-interval Metered) - A Baseline model that uses statistical sampling to estimate the electricity consumption of an Aggregated Demand Resource where interval metering is not available on the entire population.
3. Behind-The-Meter Generation - A performance evaluation methodology, used when a generation asset is located behind the Demand Resource's revenue meter, in which the Demand Reduction Value is based on the output of the generation asset. Distributed generation resources are considered "behind-the-meter" generators, such as combined heat and power (CHP) systems, wind turbines, and photovoltaic generators that generate electricity on site.

3. DRACS Design Considerations

DRACS design considerations are affected by several assumptions, dependencies, constraints, and goals. This section describes some of the considerations which need to be contemplated before writing a single line of code.

3.1 Assumptions

The following bullets include the assumptions made concerning the DRACS reference architecture:

- Compliant with DRACS Requirements (EnerNex Corporation, 2009).
- When appropriate, the system designer/implementer must be compliant with ALL appropriate standards organizations' terminology, interoperability, and messaging. This includes:
    - North America Energy Standards Board (NAESB) Demand Side Management and Energy Efficiency (DSM-EE) standards – (coming soon – NAESB is the "stakeholder" arm at FERC). Business practices to support demand side management and energy efficiency programs in the wholesale and retail electric markets.
    - NERC Demand Response reporting standards – coming soon
    - ANSI/IEEE Std 1471-2000 and ISO/IEC 42010-2007 - addresses the activities of the creation, analysis and sustainment of architectures of software-intensive systems, and the recording of such architectures in terms of architectural descriptions.
    - Open Automated Demand Response Communications Standard (OpenADR), LBNL, May, 2008
- The DRACS system is the situational understanding module within the overall DRMS system.
- Existing T&D topology data is already in existence and can be utilized by the DRACS system.
- The DRACS system includes the following high level components.
    - Relational database of Demand Resources and behavior
    - Scenario analyzer/Baseline tool
    - Simulation Model / testing tool
    - Visualization of DR topology and real time network information
- This requirements document was primarily developed for California ISO, utilities, and aggregators. However, the requirements are intentionally generic enough to apply to other states.

- Architectural design to implement System Open Architecture (SOA) based on The Open Group Architecture Framework (TOGAF).

3.2 Dependencies

Demand Response Analysis and Control System (DRACS) is dependent upon the Demand Response Management System (DRMS) for supplying information on Demand Response resources, their capabilities and attributes. This information in turn is supplied to DRMS from Demand Response Programs. DRACS receives customer usage data from Meter Data Management System (MDMS) which is supplied information on customer data from Customer Information System. DRACS receives system topology and network parameters from SCAD/EMS. SCADA/EMS provides geospatial information on network topology from Geographical Information System (GIS). GIS also provides demand resource geospatial information through Customer Information System.

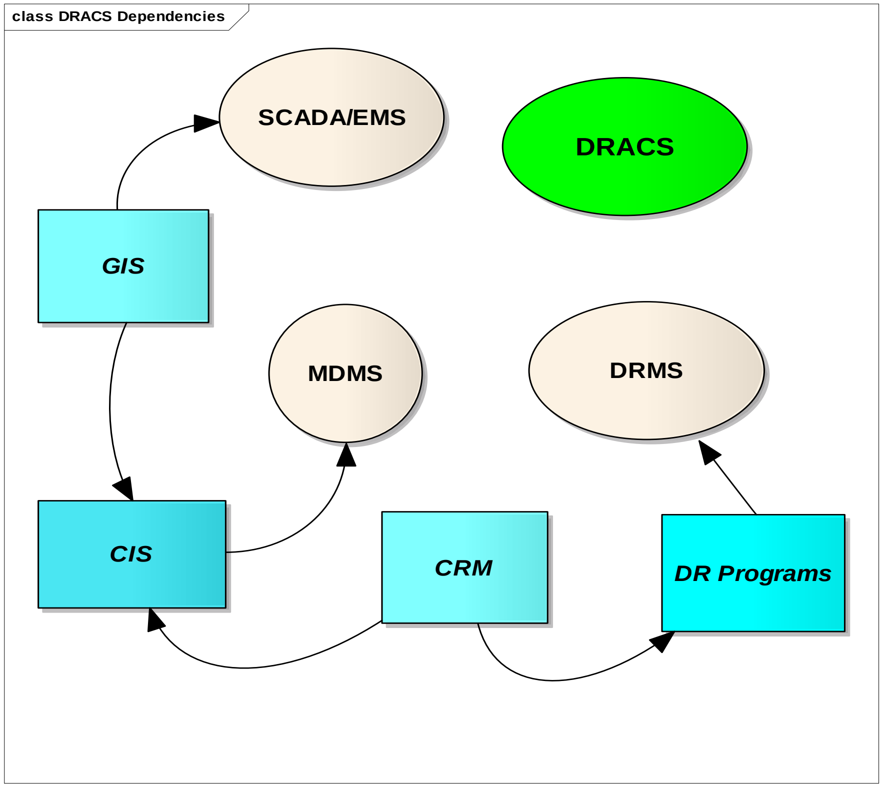


Figure 3-1: DRACS dependencies

3.3 General Constraints

- IEC 61968 Application integration at electric utilities – System interfaces for distribution management.
- IEC 61970 Application integration at electric utilities – Energy management system application program interface (EMS-API).
- Supports utility enterprise system models under development by UCAIug Open SmartGrid, AMIEnt, AMISec for common information models, information exchange models, and security.
- GridWise Architecture Council – GridWise Interoperability Context-Setting Framework

3.4 Goals and Guidelines

- Support California (CA) Demand Response programs and macro CA energy programs.
- Scalable to 50M Demand Resources.
- DRACS module and DRMS system is expected to operate within CAISO, IOUs, Energy Service Providers (ESPs), or DR Provider. Each energy management entity is responsible for managing Demand Response for its territory. Figure 3-2 is intentionally drawn hierarchically with CAISO as the responsible party for the state of CA. Each Utility is responsible for addressing CAISO DR requests. How they accomplish the DR requirements is based on their own individual business rules and Demand Resources. The same is true for the ESPs and DR Providers. They are responsible for responding to parent DR requests, but management of the DR response is at their discretion (regulatory constraints still apply, of course). In addition, IOUs may also execute a DR event to address local system constraints or reliability issues.

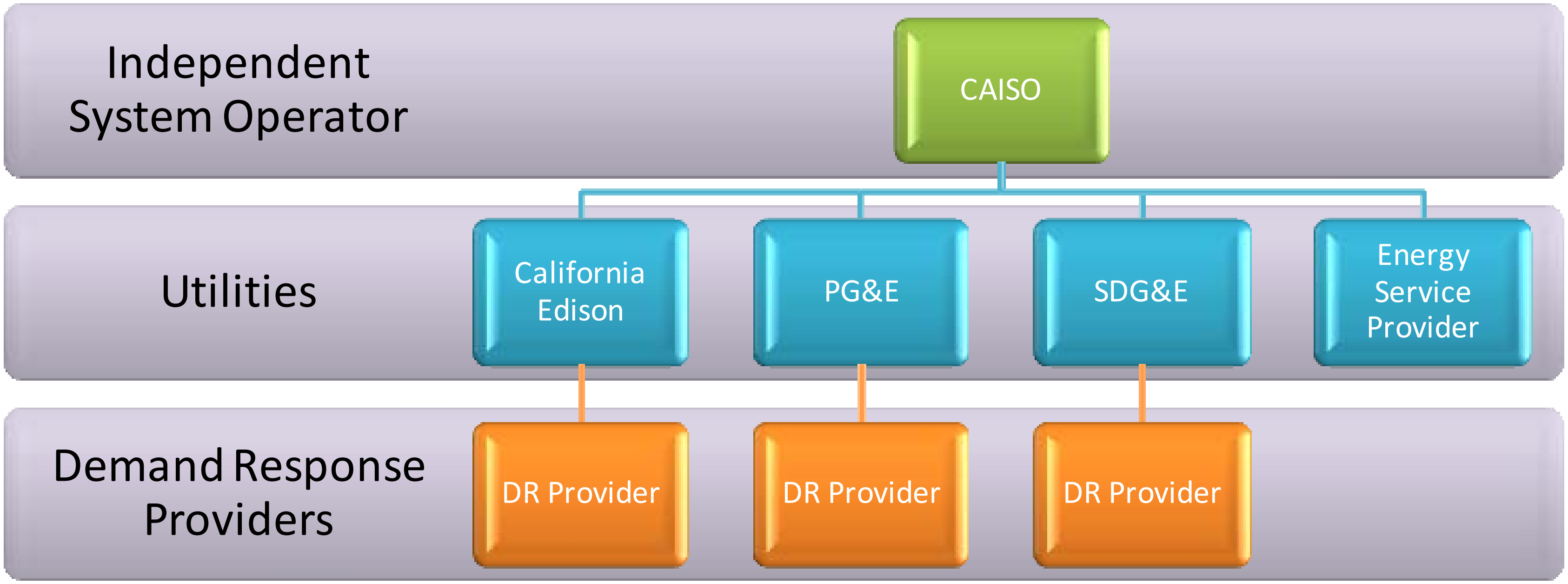


Figure 3-2: DRACS and DRMS instances will operate within CAISO, IOUs, Co-Ops, and DR Providers

3.5 Development Methods

There are many different popular and trendy software development methods, including Agile, extreme Programming (XP), and Test Driven Development (TDD). The DRACS reference architecture does not presume any specific development methodology.

Introduction

The design process begins with a goal to develop a system architecture based upon requirements distilled from use cases. It continues with a three step approach, starting with the development of a reference model specifying functional elements of the system, followed by the architectural design involving selection of standards for the functional elements, and finally the specification of a fieldable configuration involving selection of hardware and software products compatible with the predetermined standards. An implementation plan is then produced by taking into account the priorities of requirements and various constraints such as available budget and time and finally a calculation of the cost of the system is presented to management.

Current technology, along with standards and constraints are considered to be the key factors in the design for the realization of the description. Needs of users, especially of decision makers including possible future demands are reflected in the description. The user needs are determined by structured knowledge elicitation techniques using Subject Matter Experts (SMEs) workshops. Since user requirements are expected to change with time and these changes can be implemented as technology permits, the description may evolve incorporating new and enhanced features. This dynamically changing nature of the system is ensured in the design by employing evolutionary development and acquisition principle.

The system should comply with the Open System Architecture principles. Thus existing systems and systems procured in the future can be operated together enabling the system to evolve on demand smoothly. If all the system components conform to the standards, no technical interoperabiilty problems arise. Software components can easily be transported from one platform to another. Problems of vendor dependence do not exist, because any vendor who supplies products conforming to the standards can be chosen. Therefore, in a system purchase in which multiple alternatives can be evaluated using Commercial-Off-The Shelf (COTS) technology, procurement cost can be reduced.

Model Development Process Steps

The following list provides an outline of a Model Development Process. This list below can be modified and adjusted as needed. System developers can reference the numerous textbooks and industry guides that cover this subject matter in much greater detail.

Requirements Analysis	Gather information about the legacy system you're re-engineering Identify real-world domain objects Draw the domain model Do some rapid prototyping of the proposed new system Identify use cases and put them on the use case diagram Organize the use cases logically into groups. Capture this information in a package diagram Allocate functional requirements to the use cases and domain objects Write the first-draft use cases Requirements Review
Analysis and Preliminary Design	Technical Architecture Perform robustness analysis

	Disambiguate the first draft use case text Identify a first cut of objects that accomplish each scenario Update your domain model as you discover new objects and attributes Finish updating the analysis-level class diagram Preliminary Design Review
Detailed Design	Split the domain model into as many class diagrams as needed Allocate behavior by drawing sequence diagrams Generate a skeleton sequence diagram from boundary and entity objects on the robustness diagram Draw message arrows between objects Generate unit test stubs for all controls on the robustness diagram Update class diagrams with new attributes and operations Clean up the static model Review design to ensure it satisfies all the requirements Critical design review
Implementation	Generate the domain classes Coding and testing Implement unit tests Write the source code Run the tests Perform system and user-acceptance testing Code review and model update Delivery

4. Architectural Strategies

4.1 Service Oriented Architecture

DRACS and the overall DRMS system will exist within a Service Oriented Architecture (SOA). SOA architectures readily support business processes by providing application services which provide a governed, abstracted function which can be reused by many different business processes for often completely different reasons.

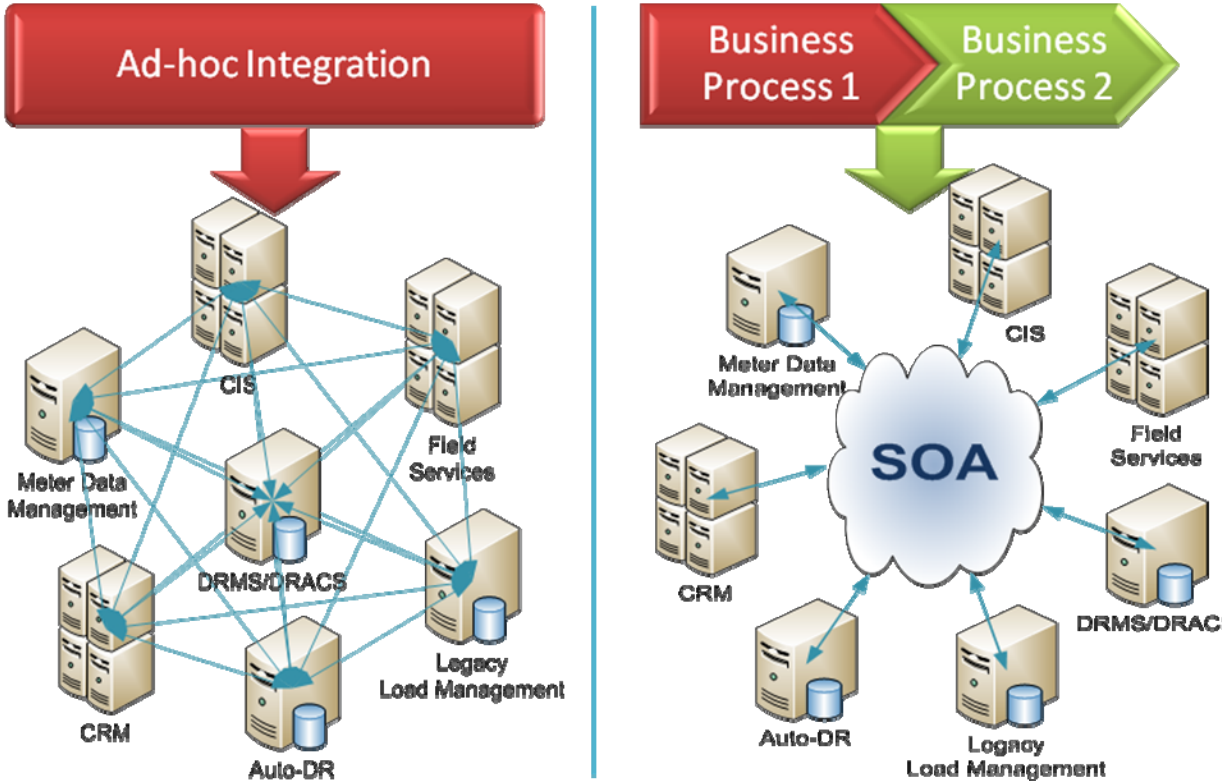


Figure 4-1: SOA Provides Business Process Flexibility by Abstracting Functionality to Services

Figure 4-1 shows direct integration from one application to another using traditional spaghetti integration side-by-side to the SOA integration philosophy where integration is provided through abstracted services that support flexibility in developing and applying different business processes in a governed SOA environment. SOA provides a “loosely coupled” integration strategy, providing flexibility and extensibility integration characteristics. SOA provides a “layered” approach, where each layer provides a level of abstraction to the layer above and below it. There are many different SOA architectural standards in the marketplace, and this document is not intended to cause a philosophical debate about the number of abstraction layers or the best SOA reference architectures. The specific SOA architecture employed is customer-dependent and will depend on the company deploying DRACS on which one is chosen. However, there are some basic layering expectations and subsequent artifacts for the DRACS SOA solution.

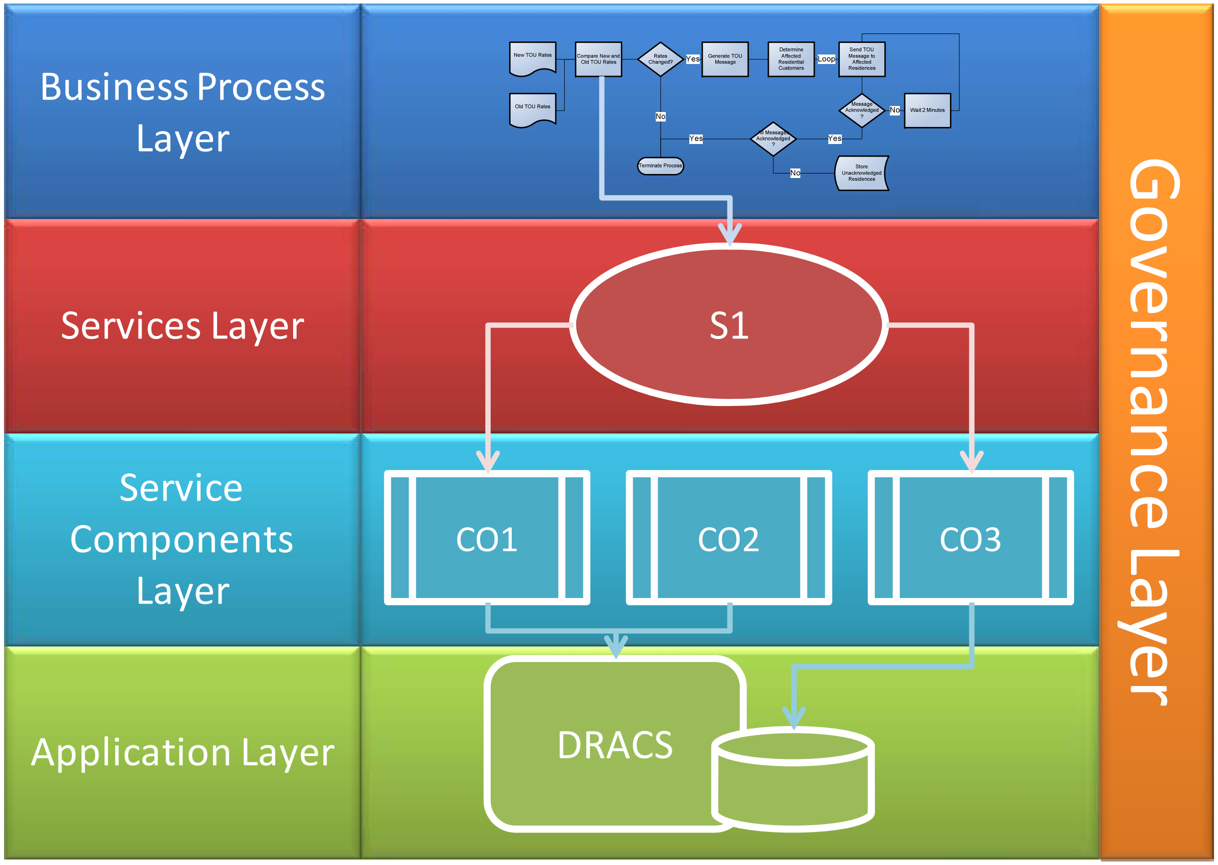


Figure 4-2: DRACS Minimal SOA Layers and Artifacts

Figure 4-2 is provided to show the minimal SOA layers and resulting software artifacts necessary in the DRACS reference architecture. At the bottom-most layer is the Application Layer where the DRACS module and the Demand Resource Attribute and Behavior database exist.

The first layer of abstraction (second level from bottom) is the Service Components Layer. The DRACS artifacts in this layer are the components. Components provide small, finite services or functions for commonly occurring activities. These services are addressable and exposed to the Services Layer. An example of a service component is a function that is passed a customer ID, and returns the Demand Resource IDs linked to that customer within the DRACS Demand Resource Attribute and Behavior database. This is normally done through web services using SOAP and XML messaging structures.

The second layer of abstraction (third level from bottom) for the purposes of this discussion is the Service Layer. The DRACS artifacts in this layer are web services. The DRACS web services provide more complex functions, such as adding or deleting a Demand Resource from the database. The web services in this layer use one or more service components from the Service Components Layer and may interact with other service components not directly related to the DRACS module, such as a mathematical calculation service component. Services may also consume other services, but this is less commonly performed at this layer and is more often performed at the Business Process Layer. Governance rules will provide guidance on how different layers and their artifacts should interact.

The final layer (topmost) of abstraction for the purposes of this discussion is the Business Process Layer. For this discussion, we will assume that this is where the service orchestration occurs and where the Enterprise Service Bus exists. Some may argue that orchestration is yet another layer, but it is not pertinent to the DRACS reference architecture or this document. Regardless, the Business Process Layer supports the combination of services from the Services Layer to support complex business process logic. An example might be when a customer removes a Demand Resource and replaces it with two new Demand Resources, where multiple services are utilized to remove the existing resource from the database and add the two new ones.

These artifacts; service components, services, and business processes, are the minimum SOA-related reference architecture requirements for DRACS.

4.2 Gridwise Architecture Council Interoperability Framework

DRACS will support the GridWise Architecture Council's *GridWise Interoperability Context Setting Framework.* The GridWise® Architecture Council's (GWAC) stated mission is to enable interoperability among the many entities that interact with the electric power system. Interoperability is defined as the capability of two or more networks, systems, devices, applications, or components to exchange information between them and to use the information so exchanged. As a step in the direction of enabling interoperability, the GWAC has proposed a context-setting framework to organize concepts and terminology so that interoperability issues can be identified and debated, improvements to address issues articulated, and actions prioritized and coordinated across the electric power community.

A context-setting framework, involves a high, organizational level community of stakeholders addressing concerns related to integrating parts of a large, complex system. A framework sits at a broad, conceptual level and provides context for more detailed technical aspects of interoperability. A model (or architecture) identifies a particular problem space and defines a technology-independent analysis of requirements. Mapping model requirements into a particular family of solutions based upon standards and technical approaches results in a *design*. A solution manifests a design into a particular vendor software technology, ensuring adherence to designs, models, and frameworks.

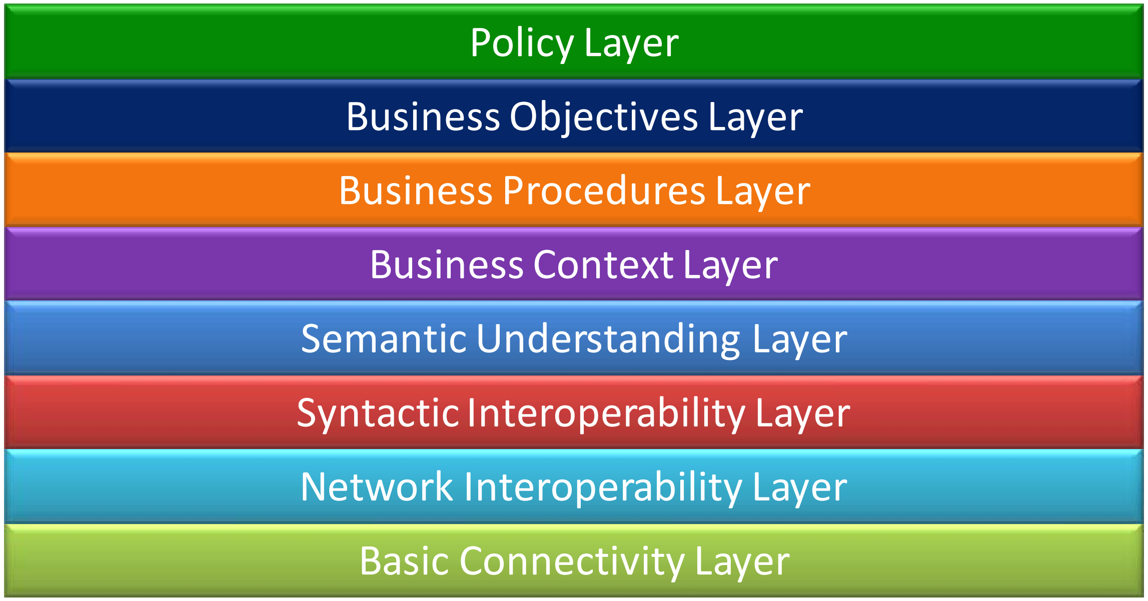


Figure 4-3: GWAC Stack Framework

The GWAC framework recognizes that interoperability is achieved when agreement is reached across many layers that span the details of the technology involved to link systems together, to the understanding of information exchanged, to the business processes and organizational objectives that are represented in business, economic, and regulatory policy.

The framework will be used to identify the key interoperability issue areas and can help resolve interdependencies within the electric system and with other infrastructures. The framework also enables the representation and exchange of ideas, aligning, and harmonizing technical approaches with accompanying management procedures and business processes.

5. DRACS Reference Architecture and Design

This section provides a discussion on the DRACS high level reference architecture and more detailed reference architecture for each of the primary DRACS components (Database, Demand Resource Situational Understanding Engine, and Visualizaton) and the database loading/maintenance utilities.

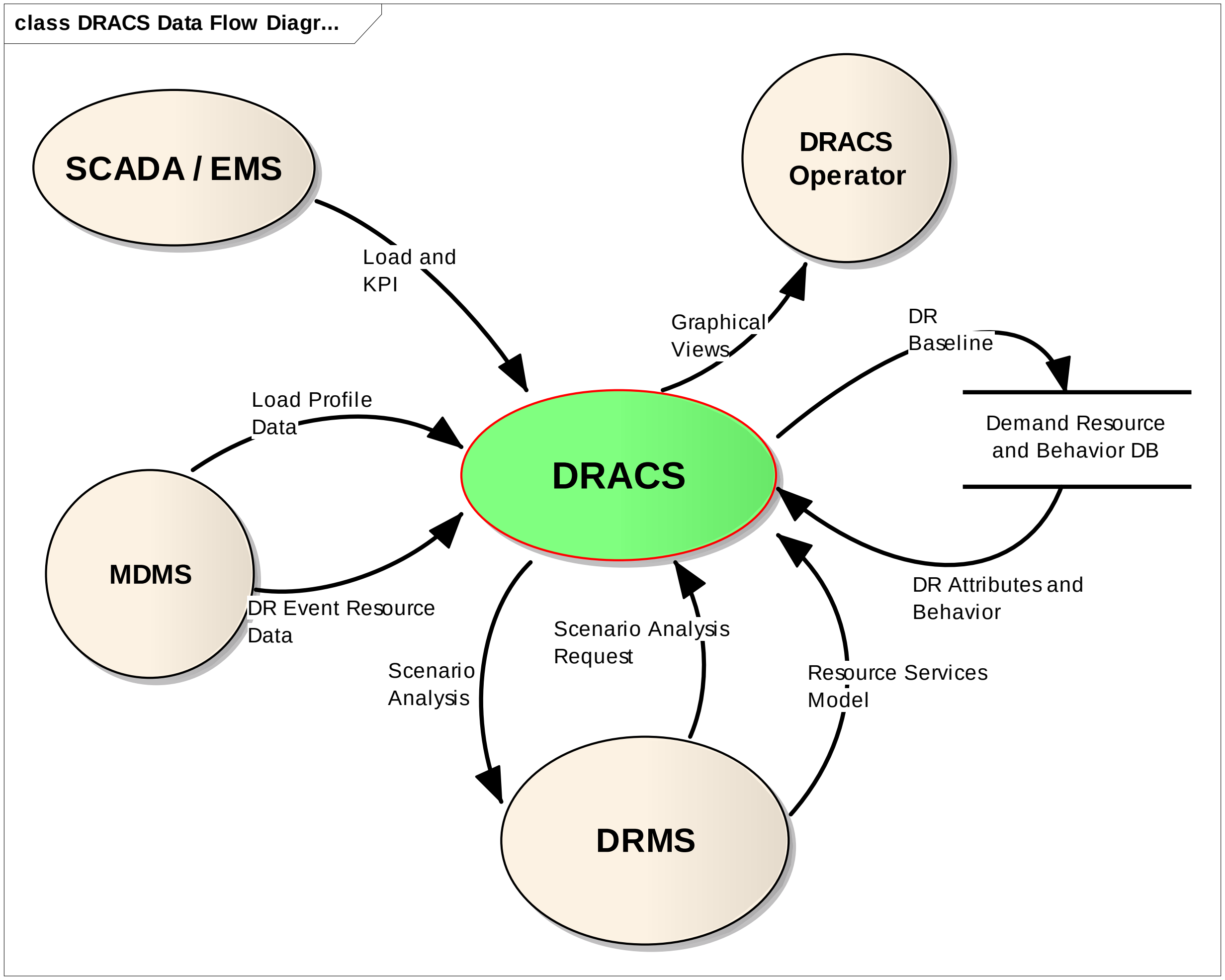


Figure 5-1: DRACS Data Flow

Figure 5-1 shows the DRACS data flow and messages between other Enterprise systems. There are four (4) primary external and one (1) internal actors/systems DRACS interacts with:

1. DRACS operator – the DRACS operator is an external actor that is responsible for the overall operation of the DRMS/DRACS system and the "situational understanding" of the Demand Response topology. He interacts with DRACS through the graphical visualization component, overseeing DR events, and observing the DR network topology in real time.
2. Demand Resources and Behavior Database – this relational database provides the data storage and retrieval needs for participants, Demand Resources, Demand Response programs, DR events, and DR systems. DRACS also provides Baseline calculation information for each Demand Resource, which is stored in the database.
3. DRMS – DRACS is a module within DRMS. Other modules, components, and routines within DRMS develop scenarios and pass them to DRACS for analysis. DRACS provides scenario confidence analysis in return. The DRMS also is responsible for creating Demand Response Programs and assigning participants, which is stored in the Demand Resource Attribute and Behavior database.
4. Meter Data Management System (MDMS) – The MDMS is part of the Advanced Metering Infrastructure (AMI) external system that captures and stores meter data traffic. Although Demand Response messaging standards are currently being defined, those messages will eventually be routed through the MDMS where DRACS can retrieve Demand Resource attribute and event behavior data.
5. SCADA/EMS – the SCADA/EMS is an external system that provides DRACS with real time load and key performance indicator (KPI) data on the current state of the network.

5.1 Database

DRACS requires a relational database to house participant and Demand Resource information. DRACS and DRMS are the only systems that should have direct access to the database. As discussed in Section 4.1, Service Oriented Architecture, other systems requiring access to the DRACS database will do so through an abstracted SOA service layer in order to control access and reduce opportunities for data corruption. Database scripts, stored procedures (e.g. summation of Demand Resource Baselines for a single participant), and data are not directly accessible to any programs outside of the DRMS system. Database SOA services require authentication to access information within the DRACS database.

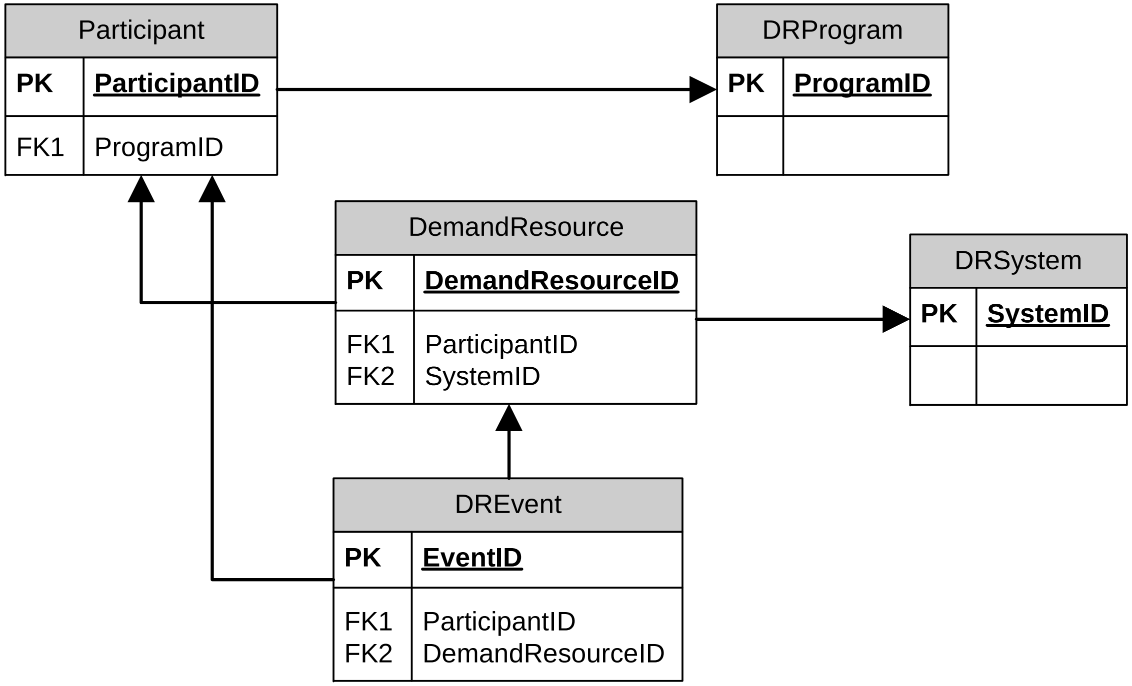


Figure 5-2: DRACS database has 5 primary entities

There are five primary database entities which need to be linked within the database:

1. Participant data. This includes pertinent information such as customer name, historic Demand Response event behavior, circuit details, and address. Participants may include zero or more Demand Resources, include zero or more Demand Response Events, and may be a member of zero or more Demand Response Programs.
2. System and Communication data. System and communication data includes the various Demand Response systems and communication mechanisms available for dispatch. For instance, these systems may include AMI, existing Load Control programs, or the newer Auto-DR systems. This information is used to help direct communication traffic to the affected Demand Resources in a DR event. Demand Response Systems include one or more Demand Resources.
3. Demand Resource data. This data provides details about demand response equipment, including device type, manufacturer, installation details, location, and Baseline. Demand Resource entities are members of exactly one Participant, are members of one or more Demand Response System, and include zero or more Demand Response Events.

4. Demand Response Program data. DR Program data includes information about the organization's various participant programs for managing load through Demand Response. DR Programs include zero or more participants.
5. Demand Response Event data. This data provides details about past events, including initiating organization, business rule(s) invoked, load management objectives, load management results, date, time, and duration. DR Events include zero or more Participants, and include zero or more Demand Resources.

Lawrence Berkeley National Laboratory (LBNL) has developed a data model for their Auto-DR Demand Response Automation Server (DRAS) technology that includes Entity elements for Participants, DR Programs, and DR Events that are similar to those elements needed in the concrete DRACS architecture. Although DRAS and DRACS are dramatically different architectures with different operational objectives, the underlying database entities are quite similar, and provide a good start for designing the concrete DRACS data model. DRAS data model details are available by contacting LBNL directly.

5.2 Demand Resource Situational Understanding Engine

There are actually two Demand Resource Situational Understanding Engines in the DRMS system. The first one is the operational system that is integrated within DRACS, and is the one discussed in this section. The operational system uses real time data from the SCADA/EMS system, behavioral data from the database, and internal algorithms to predict actual responses from the Demand Resources identified within a specific DR event scenario. More on this will be discussed later in this section.

The second Demand Resource Situational Understanding Engine is a separate simulation modeling tool which uses worst case historical data on communication latencies, Demand Resource responses, and DR topology complexity to predict potential system instabilities and offer potential solutions. The off-line simulator is used for planning purposes and performs analysis off-line, separate from the operational DRACS system. As an example, the DRMS issues a DR event command to a diverse set of Demand Resources where 30% of them are located remotely with poor SCADA coverage and slower communications. The DRMS real time monitoring determines that the system is not meeting the response demands of the DR scenario, when in fact, the Demand Resources are responding appropriately, but the communication delay causes the DRMS to create a scenario with an increase in the number of affected Demand resources and send another dispatch signal out across the system. When the next SCADA update occurs, the DRMS realizes it has actually overshot the DR event requirements and is forced to decrease the Demand Resource and issue another DR signal. This potential real-time system instability is the reason an off-line simulation modeling capability is necessary. For more details on the off-line DRACS simulator, refer to the Scenario Analyzer and Simulator Requirements for Demand Response Analysis and Control System document (EnerNex and Oak Ridge National Laboratory, 2009).

The operational Demand Resource Situational Understanding Engine is the "heart" of the real time DRACS system. This DRACS component is highlighted in the block diagram in Figure 5-3, below.

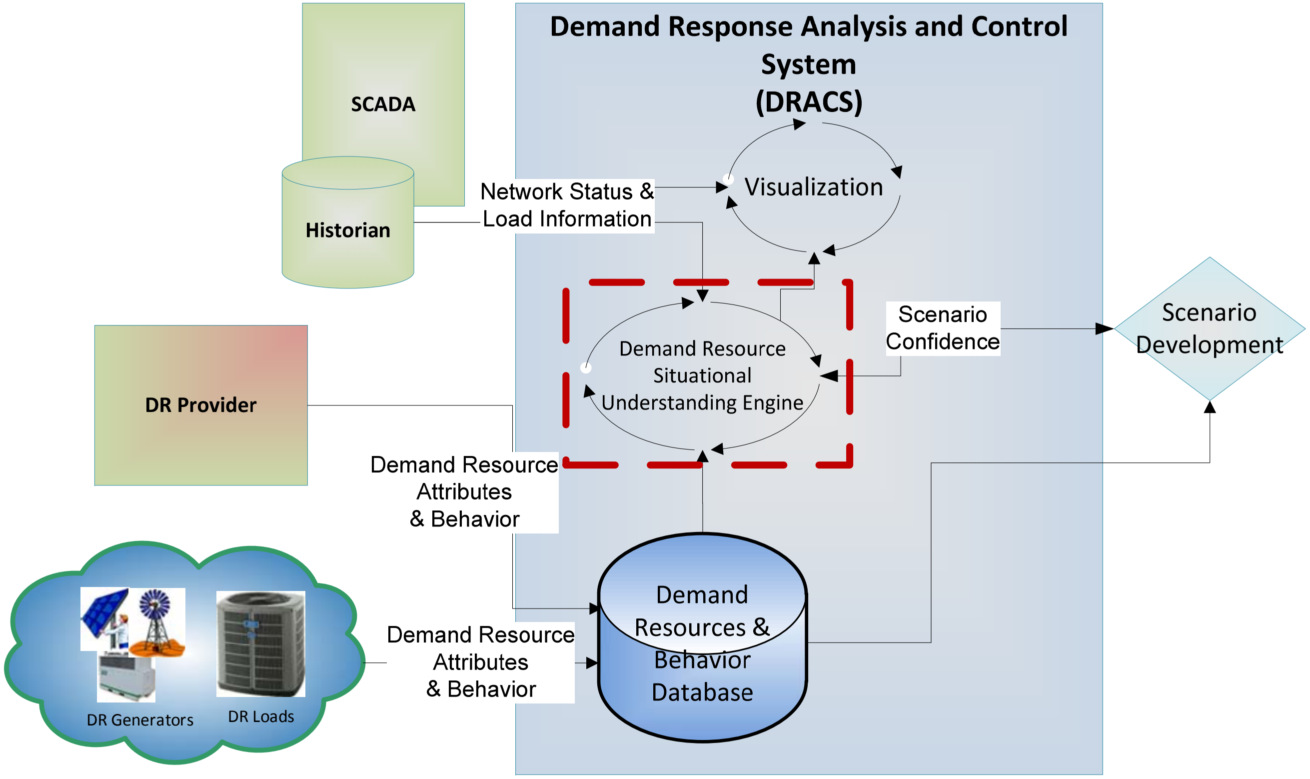


Figure 5-3: DRACS Demand Resource Situational Understanding Engine component highlighted

The operational Demand Resource Situational Understanding Engine is composed of several subcomponents. Figure 5-4 provides a block diagram and data flow of the operational scenario analyzer. The subcomponents include:

1. Database connectors to Demand Resources and Behavior Database
2. SCADA/Historian interface
3. Scenario ingest module
4. Dispatch prioritization module
5. Power flow solver
6. Contingency prioritization module
7. Human-user alternatives proposer and visualization interface
8. Tracking module for dynamic state-estimation
9. Dispatch performance tracker

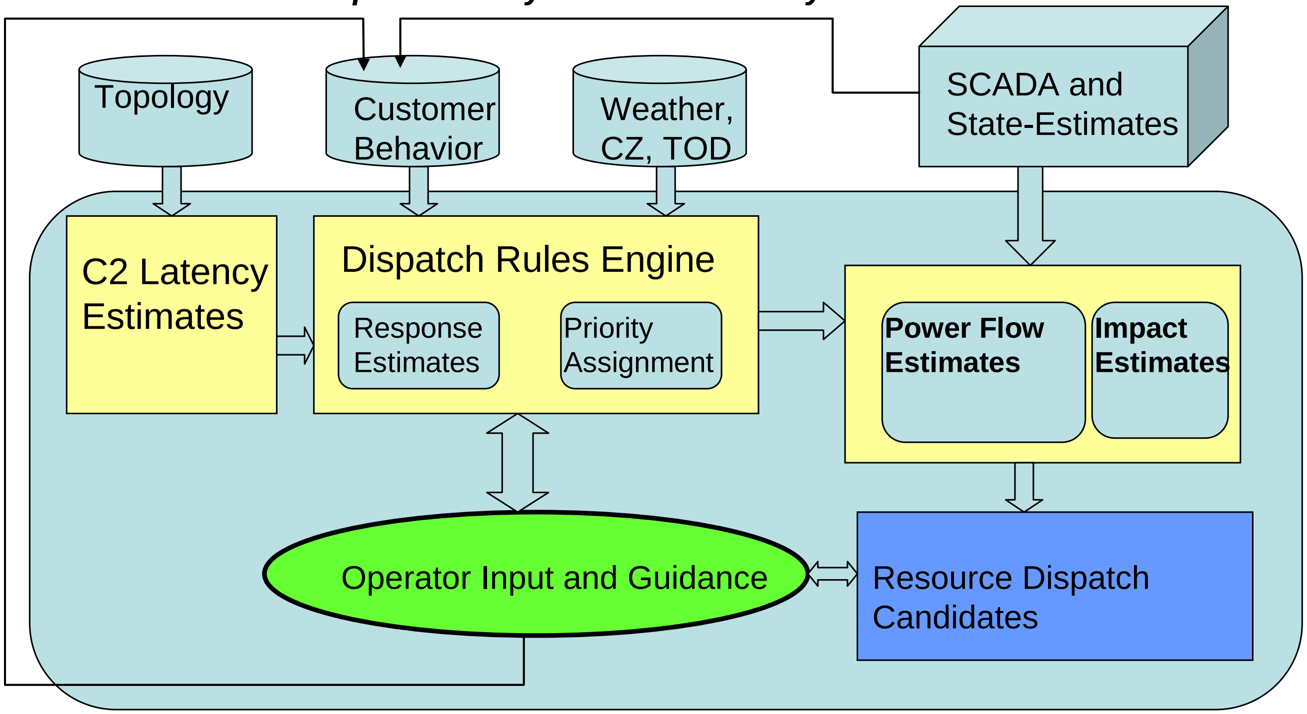
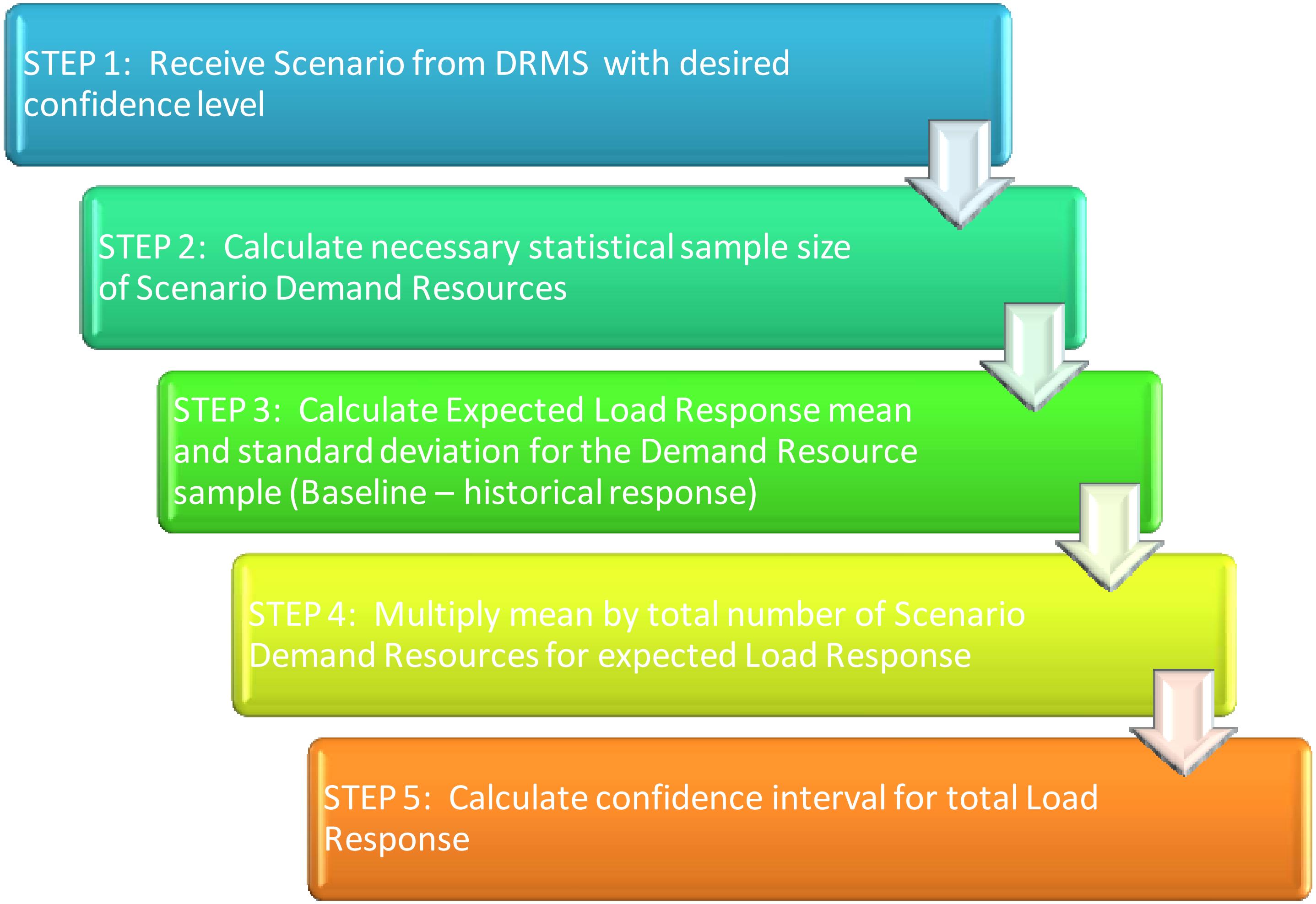


Figure 5-4: Operational Demand Resource Situational Understanding Engine Subcomponent Block Diagram

One of the primary objectives of the operational system is to provide scenario confidence intervals for meeting DR event requirements in real time. Figure 5-5 shows the steps for creating Baseline Confidence Intervals for evaluating Demand Response scenarios. DRACS receives a scenario, population total, and expected *Confidence Level* from other modules within the DRMS. DRACS determines the statistically-significant sample size. Next it calculates or retrieves Baseline information for the sample set and calculates the expected Load Response based on historical or empirical data. The expected Load Response calculation is the "secret sauce" and the reason DRACS requires a true "situational understanding" of the Demand Response landscape. It will use historical data, current electrical network status, weather, social response algorithms, communication loss and latencies, and any number of other proprietary algorithms to accurately predict the expected Load Response for the scenario. DRACS uses all this information to calculate the expected Load Response mean and standard deviations. The final step returns the total expected Load Response and *Confidence Interval* to the requesting module in DRMS.



STEP 1: Receive Scenario from DRMS with desired confidence level

STEP 2: Calculate necessary statistical sample size of Scenario Demand Resources

STEP 3: Calculate Expected Load Response mean and standard deviation for the Demand Resource sample (Baseline – historical response)

STEP 4: Multiply mean by total number of Scenario Demand Resources for expected Load Response

STEP 5: Calculate confidence interval for total Load Response

Figure 5-5: DRACS computes confidence intervals using statistical methodologies in evaluating scenarios

When calculating scenario confidence information, some basic statistical concepts need to be understood:

- Confidence Level. The confidence level is the amount of acceptable *uncertainty*. Typically, C*onfidence Levels* used in statistic calculations are set at 95% or 99%, which correspond to 5% and 1% uncertainty, respectively. It will likely take some experimentation and simulation to determine optimal C*onfidence Levels* in scenario analysis.

- Confidence Interval. Confidence intervals are the potential amount of *error* in the statistical calculation. If the C*onfidence Interval* is 95%, then the *Margin of Error* is 100% - 95%, or 5%. The *Confidence Interval* represents a range of values, consistent with the data set, for which there is a statistical probability (or confidence), that it contains the "true" value. This parameter is commonly set at 95%.
- Sample Size. The *Sample Size* is the minimum size of the survey. This smaller sampling of the overall population provides a statistically accurate representation of the total population. The higher the *Confidence Level,* the larger the S*ample Size*, and the longer it takes to compute *Confidence Interval* results.

5.3 Visualization

The visualization capabilities of DRACS are two-fold. First, DRACS provides a visual representation of the overall demand response topology. This entails a hybrid visualization of the grid, streets, terrain, landmarks, weather, and demand resources, such as depicted in Figure 5-6. This screen shot graphic is a visualization of the Southern California grid by a product under development by Oak Ridge National Laboratory (ORNL) using the Google Earth Application Programming Interface (API).

DRACS visualization of the demand response topology also needs to include the participants and Demand Resources within the DRACS operator’s territory. The map will depict the network topology with multiple layers that can be switched on or off, including those for terrain, Demand Resources, and participants. The overall visualization requirements are to show where the energy is, where it is being used, and where it is needed in order to support the situational understanding of the Demand Response topology. Icon shape and relative size describe the type of participant and Demand Resource and their Baseline or generating capabilities. A search tool allows the operator to filter participants or Demand Resource assets by asset class, size of load management capability, location, service type, Demand Response program, Demand Response system, and Demand Response event.

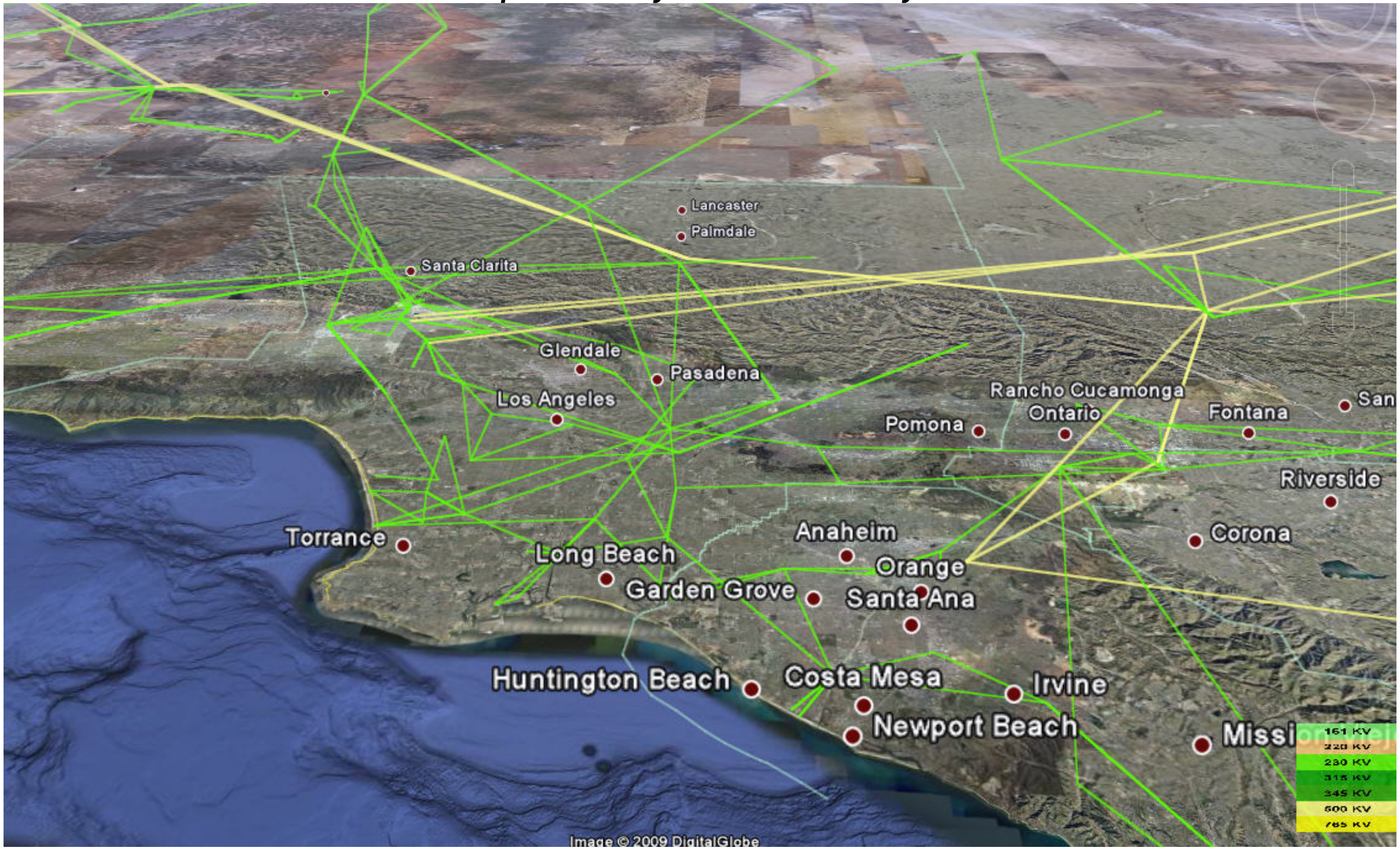


Figure 5-6: Oak Ridge National Laboratory grid visualization using Google Earth

The second visualization capability of DRACS is in providing real time distribution network topology information visualization and key performance indicators (KPI). Loads, temperature and other weather data, outages, congestion, voltage loss, and other pertinent information which can affect the ability to support a DR request are shown as more switchable layers on the topology map or as separate instrumentation "panels" on a separate screen. A successful DRACS visualization tool merges the demand response topology and the pertinent real time data.

Figure 5-7, below, is an image from the OSISoft Boulder City Smart Grid Project briefing presented at Grid-Interop 2008 by Lee Margaret Ayers. This project is a consortium partnership between Xcel Energy, Accenture, Current Group, GridPoint, Schweitzer Engineering Laboratories, SmartSynch, Ventyx, and OSISoft. The concept was developed with the intention of providing an international showcase around a small city with a fully integrated smart grid solution. One of the primary objectives of the project was to have a clear understanding of where the energy is and where it is actually needed. Figure 5-5 shows the OSISoft visualization of energy and solar usage footprints. This is representative of the DRACS visualization needs for customers and Demand Resources.

(Lee Margaret Ayers, OSISoft, 2008)

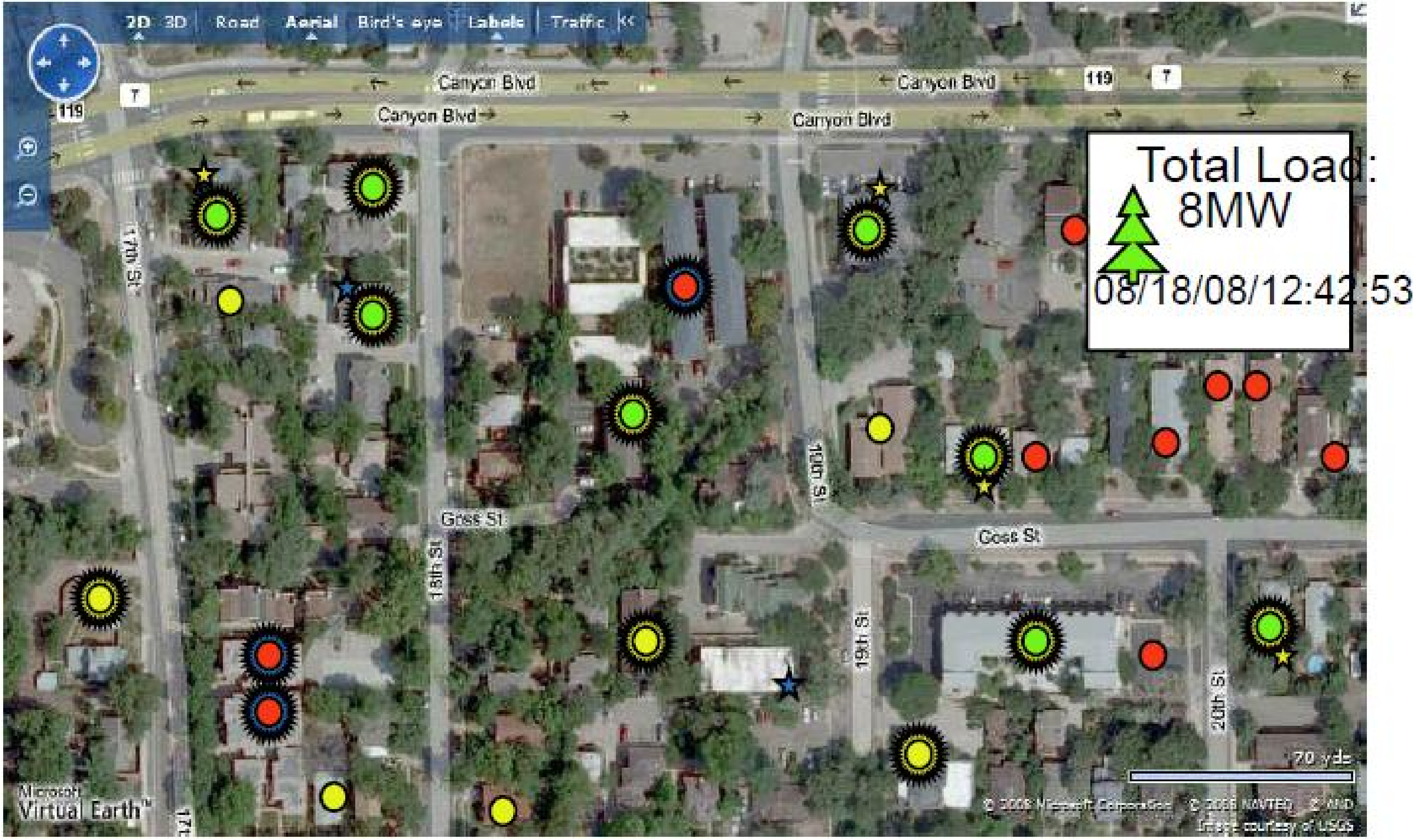
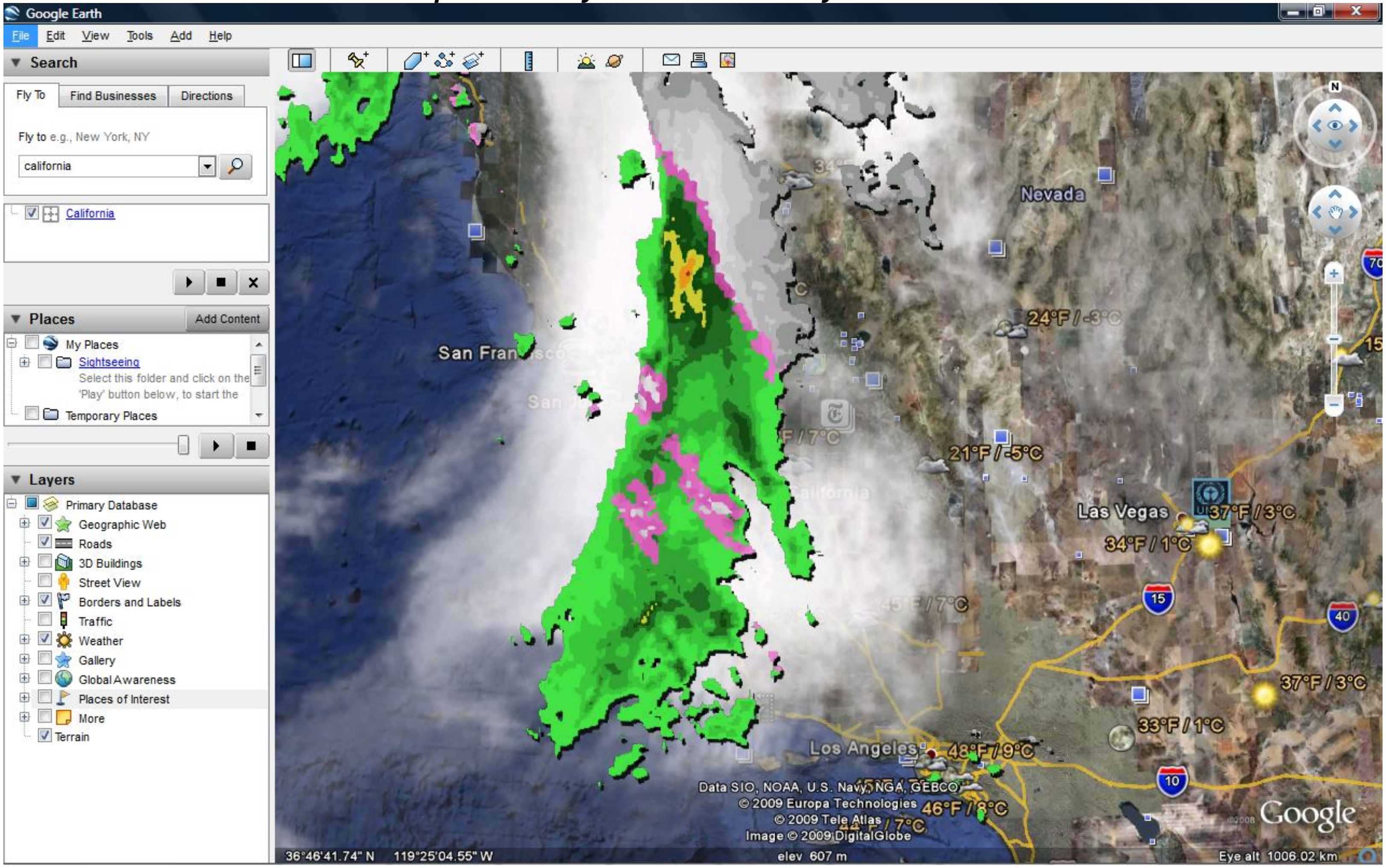


Figure 5-7: DRACS shows where the energy is, where it is being used, and where it is needed

As Demand Response events are initiated, both audible and visual alerts will notify the operator of the event. The visualization tool will show the operator which participants and Demand Resources are affected, the target loads, and the actual loads (at whatever fidelity is available) at metered locations within the event participant territory.

Although an integrated visualization solution with topology map and KPIs on one screen is interesting, it is not practical to assume everything will be displayed at once. Incorporating that much information on a single screen will be cluttered and is simply too much information for the human brain to process. The important concept to keep in mind with the topology map is the fact that the operator needs to understand where the energy is, where it is being used, and where it is needed. This provides the "situational understanding" feedback to the operator showing whether the system is working within tolerances or not. Clicking or mousing over icons should instantiate numeric displays that provide details about what that icon represents. Numeric load information may be displayed based on the operator's discretion at aggregation points where sensors provide that information. These load aggregation features are switchable load hierarchies within the DRACS visualization environment, beginning at the individual metered load, and summing successively up through distribution transformer, to branch circuit, to distribution feeder on up to distribution substation. Local subtransmission summaries on up to regional transmission system views should also be possible. In fact, the operator should have as much control as programmatically possible in deciding what is and what is not displayed and at what fidelity that information is displayed. KPIs, Demand Resources, participants, weather, and system loads are switchable layers in the visualization component.

Figure 5-8 shows Google Earth's ability to display switchable layers for various features it makes available to the user. This familiar Graphical User Interface (GUI) provides a menu across the top and search/feature tools on the left. The feature tool includes a layer-switching mechanism that provides checkable features in a hierarchical structure that can easily be turned on or off. This capability is similarly necessary with DRACS in order to provide the operator with the visualization customization to allow him to observe and make decisions based on the resulting information.



Fly to: california

☑ California

☐ My Places
☐ Sightseeing
Select this folder and click on the 'Play' button below, to start the
☐ Temporary Places

■ Primary Database
- ☑ Geographic Web
- ☑ Roads
- ☐ 3D Buildings
- ☐ Street View
- ☑ Borders and Labels
- ☐ Traffic
- ☑ Weather
- ☐ Gallery
- ☐ Global Awareness
- ☐ Places of Interest
- ☐ More
- ☑ Terrain

Figure 5-8: Google Earth provides switchable layers similar to DRACS visualization operator options

5.4 Demand Resource Attributes and Behavior Collection

As Demand Resource populations increase and the DR landscape and standards mature, collection of attribute and behavior information will also mature and mechanisms will be invented for ensuring timely, standardized, automated ways for maintaining the database of resources. At this point in time, it is premature to develop a reference architecture that reflects that level of maturity and automation until such standards and infrastructure exist. It is important to acknowledge that the future mechanisms for attribute and behavior data collection is the least understood area for the DRACS reference architecture at the time of writing this document. However, it is also the most concerning and should be the number one priority for improvement. Opportunities for standards development around asynchronous self-announcement/destruction, self-description, attribute modifications, and DR event response behavior are important considerations in developing the DR communication and messaging standards over the next year or two. Without these Common Information Model (CIM)[1] industry standards considerations, maintenance of the DRACS Demand Resource database will become untenable as the large volume of Demand Resources expected begin entering the DR topology and the resulting scenario prediction capabilities for DRACS will become suspect. In other words, the less accurate the Demand Resource data is, the less accurate the scenario prediction capabilities of DRACS. So until new standards emerge for Demand Resource data collection, this reference architecture will address attribute and collection behavior in less automated and practical ways.

To assist in developing behavior models, DRACS will start with standard definitions developed by NAESB for Baseline.

> “A Baseline is an estimate of the electricity that would have been consumed by a Demand Resource in the absence of a Demand Response Event. The Baseline is compared to the actual metered electricity consumption during the Demand Response Event to determine the Demand Reduction Value. Depending on the type of Demand Response product or service, Baseline calculations may be performed in real-time or after-the-fact. The System Operator may offer multiple Baseline models and may assign a Demand Resource to a model based on the characteristics of the Demand Resource’s Load or allow the Demand Resource to choose a performance evaluation model consistent with its load characteristics from a predefined list. A baseline model is the simple or complex mathematical relationship found to exist between Baseline Window demand readings and Independent Variables. A baseline model is used to derive the Baseline Adjustments which are part of the Baseline, which in turn is used to compute the Demand Reduction Value. Independent variable is a parameter that is expected to change regularly and have a measureable impact on demand.”

---

[1] The Common Information Model is part of the IEC 61968 Application integration at electric utilities. This standard defines system interfaces for distribution management and provides guidelines and directives for messaging and interoperability between applications, devices, and systems.

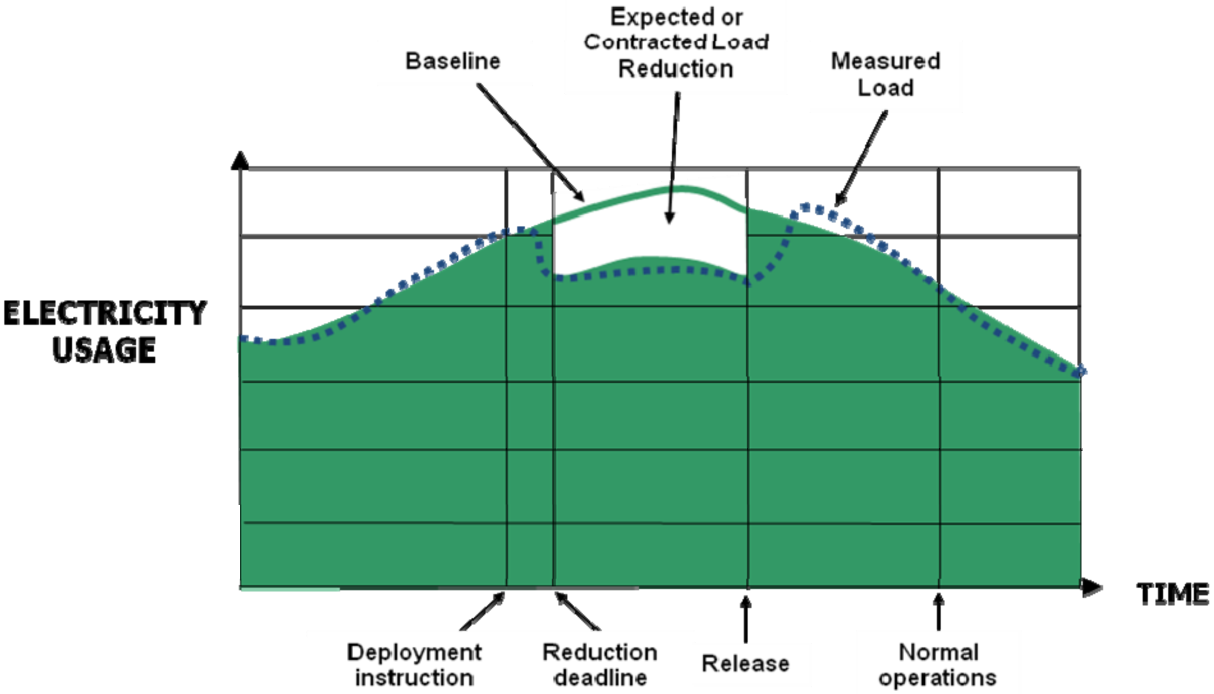


Illustration of Baseline Concept

6. Policies and Tactics

6.1 Enterprise Service Bus

The big three California Investor Owned Utilities (IOUs) , Pacific Gas & Electric (PG&E), California Edison, and San Diego Gas & Electric (SDG&E), incorporate Enterprise Service Buses (ESBs) as a tactic in supporting the SOA strategy. In other words, ESBs are the tactical solution for managing and orchestrating the SOA architectural strategy.

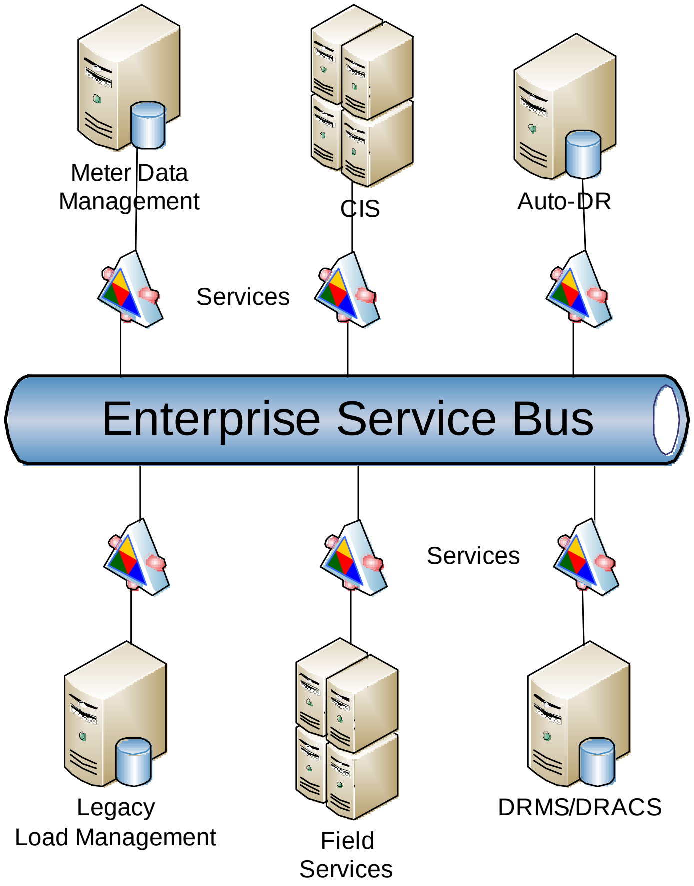


Figure 6-1: ESB provides tactical solution for SOA strategy

ESBs are a form of message-oriented middleware that act as a broker in managing and combining messages from multiple applications and systems within the Enterprise. Applications within the Enterprise communicate through the ESB. The ESB supports this communication through standards and scripting, providing a means for stringing services together as business processes. The ESB provides orchestration of these services and business processes in a structured and automated environment. ESBs support multiple transport mechanisms, routing communication between services, mediation, business process orchestration, and quality of service support including security (authentication and encryption services, message signing), message delivery reliability, and managing all the messaging transactions. The ESB also provides auditing, logging, and monitoring. Some systems may also include analytical extensions that monitor and report business activity across the bussing system.

6.2 GridWise Architecture Council

The Gridwise Architecture Council's Interoperability Context-Setting Framework is a foundation for setting up a process to bring awareness to interoperability issues and eventually establish an interoperability culture. The framework and it associated materials will be used for the advancement of the interoperability of the electric system.

The goal of GWAC is to promote a concept called interoperability, which incorporates the following characteristics:

- The exchange of meaningful, actionable information between two or more systems across organizational boundaries
- A shared understanding of the exchanged information
- An agreed expectation for the response to the information exchange
- A requisite quality of service: reliability, fidelity, and security.

GWAC System Integration Philosophy

GWAC's emphasis is on easing the task (reducing the distance to integrate) of those who integrate and configure automation components into the system. When considering interacting automation components that are managed by different organizations and departments. the transacting parties clearly and formally establish the lines of authority and rules of engagement, but maintain their autonomy while collaborating to share their resources in a federated manner. This system integration philosophy examines the following areas:

- Agreement at the Interface—A Contract
- Boundary of Authority
- Decision Making in Very Large Networks
- The Role of Standards

High Level Categorization

The GridWise has identified eight interoperability categories that are relevant to the mission of systems integration and interoperation in the electrical end-use, generation, transmission, and distribution industries. These fall into the following major categories: technical, informational, and organizational. The organizational categories emphasize the pragmatic aspects of interoperation and represent the policy and business drivers for interactions. The informational categories emphasize the semantic aspects of interoperation and focus on what information is being exchanged and its meaning. The technical categories emphasize the syntax or format of the information and focus on how information is represented within a message exchange and on the communications medium.

- Technical Aspects
  - Category 1: Basic Connectivity
  - Category 2: Network Interoperability
  - Category 3: Syntactic Interoperability
- Informational Aspects
  - Category 4: Semantic Understanding
  - Category 5: Business Context
- Organizational Aspects
  - Category 6: Business Procedures
  - Category 7: Business Objectives
  - Category 8: Economic and Regulatory

Cross-cutting issues are areas that the GWAC framework addresses in order to achieve interoperation. These usually are relevant to more than one interoperability category of the framework. Topics related to Cross-cutting issues are listed as follows:

1) Shared Meaning of Content
2) Resource Identification
3) Time Synchronization and Sequencing
4) Security and Privacy
5) Logging and Auditing
6) Transaction and State Management
7) System Preservation
8) Quality of Service
9) Discovery and Configuration
10) System Evolution and Scalability

7. Glossary

Glossary	
Term	**Definition**
Architecture	The fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.
Baseline	A Baseline is a method of estimating the electricity that would have been consumed by a Demand Resource in the absence of a Demand Response Event. The Baseline is compared to the actual metered electricity consumption during the Demand Response Event to determine the Demand Reduction Value. Depending on the type of Demand Response product or service, Baseline calculations may be performed in real-time or after-the-fact.
Capacity Service	A type of Demand Response service in which Demand Resources are obligated over a defined period of time to be available to provide Demand Response upon deployment by the System Operator.
Demand Resource	A Load or aggregation of Loads capable of measurably and verifiably providing Demand Response.
Demand Resource Service	Demand Resources can support four (4) different types of services; Energy, Reserve, Capacity, or Regulation service
Demand Response	A temporary change in electricity consumption by a Demand Resource in response to market or reliability conditions. For purposes of this requirements document, Demand Response does not include energy efficiency or permanent Load reduction.
Demand Response Provider	Also called an energy aggregator or retail energy provider. The DR provider manages its own set of demand resources for economic incentives and rewards, usually from a utility or ISO, but could also be another DR Provider.
Energy Service	A type of Demand Response service in which Demand Resources are compensated based solely on Demand reduction performance.
Regulation Service	A type of Demand Response service in which a Demand Resource increases and decreases Load in response to real-time signals from the System Operator. Demand Resources providing Regulation Service are subject to dispatch continuously during a commitment period. Provision of Regulation Service does not correlate to Demand Response Event timelines, deadlines and durations.
Reserve Service	A type of Demand Response service in which Demand Resources are obligated to be available to provide

	Demand reduction upon deployment by the System Operator, based on reserve capacity requirements that are established to meet applicable reliability standards.
System Operator	A System Operator is a Balancing Authority, Transmission Operator, or Reliability Coordinator whose responsibility is to monitor and control an electric system in real time (based on NERC definition). The System Operator is responsible for initiating Advance Notifications, Deployment, and Release/Recall instructions.

8. Bibliography

EnerNex Corporation. (2009, January 20). DRACS Requirements FINAL V1.3.doc. Knoxville, TN, USA.

Lee Margaret Ayers, OSISoft. (2008, November 11). Business and Policy, Utility Smart Grid Strategies Briefing to Grid-Interop 2008. San Leandro, CA, USA.

Organization for the Advancement of Structured Information Standards (OASIS). (2008, April 23). *Service Oriented Architecture Reference Architecture.* Retrieved January 26, 2009, from OASIS: http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/