Impact of Secure, Scalable Performance on Demand Response Communication Architecture

Dave Hardin, EnerNOC
Boston, MA
dhardin@enernoc.com

Scott Neumann, UISOL
Ramsey, MN
sneumann@uisol.com

Keywords: demand response, communications

Abstract

Economic viability is critical for demand response (DR) service providers. One factor that impacts economic viability is the capability to scale while maintaining high performance. Minimizing the costs associated with supporting large numbers of homes and facilities within a range of demand management programs is a key requirement when developing system architectures and selecting communications technology.

This paper will describe how quality-of-service (i.e. non-functional) requirements affect communication system architecture and why the selection of high performance, secure communications technology can provide a path toward greater interoperability by minimizing the diversity of technology required to satisfy a wide range of communication performance requirements when interfacing with large numbers of customer homes and facilities.

1. OVERVIEW

Electric systems are highly-interconnected complex real-time networks of electrical energy flowing from sources of generation to loads of all sizes and characteristics over large geographical regions. The reliable flow of energy requires that system voltage, phase and frequency remain stable over both very short time periods and long time periods as planned changes and unplanned perturbations occur in generation, circuits and loads. In order for demand-side resources to be applied to offset the effects of these perturbations, they must be able to respond within the time period required to maintain closed-loop stability. In other words, they need to respond in real-time. The inability to respond fast enough can result in loop oscillations and system instability.

Energy consuming systems within demand-side entities such as homes and C&I (commercial and industrial) facilities vary in their ability to alter their energy consumption in a timely fashion. Some systems can respond fast, such as electric hot water heaters, while others respond slowly. This wide range of time responsiveness permits a home or facility to participate in more than one type of demand response program. As an example, a facility may participate simultaneously in a spinning-reserves ancillary service market as well as a day-ahead energy market where both are served by the same service provider. [1]

In the context of modern communication systems, the 20th century design principle “form follows function” rings true but can be restated as “system architecture follows system requirements”.

The integration of distributed energy resources including automated demand response requires that communications infrastructure and protocols are capable of handling a wide range of system performance requirements.

2. THE ROLE OF FUNCTIONAL AND QUALITY OF SERVICE REQUIREMENTS ON SYSTEM ARCHITECTURE

Without diving into formal architectural frameworks such as TOGAF[1], system architecture essentially refers to the model that defines the structure and behavior of a system from different viewpoints. It describes how the system components fit together to achieve the goals and objectives of the system. System architecture is driven by the business requirements because they scope the system and drive technical design.

A typical architectural framework provides a set of layered system views with varying degrees of abstraction from high-level system design to physical system design. These layers may include:

- Architecture Vision
- Business Architecture
- Information Systems Architecture
    - Data Architecture
    - Application Architecture
    - Communication Architecture
    - Security Architecture
- Technology Architecture
- Technical Reference Model
- Detailed Platform Taxonomy

The focus of this paper is communication architecture.

[1] http://www.opengroup.org/togaf/

System requirements include both functional and quality of service (QoS) requirements. Functional requirements describe specific behaviors or functions that a system needs to perform and qualities-of-service requirements describe the constraints associated with the behaviors or functions such as security, performance, latency, supportability, maintainability, availability, and scalability. Functions typically define the components and interaction patterns of a system and are defined by use cases (e.g. DR event, meter telemetry services). Qualities of service requirements further define and clarify the characteristics of the components and interactions and thus constrain the physical design of the system. In a services oriented system, functional requirements will define the service signatures, payloads and interaction patterns and quality-of-service requirements will drive the security and communication architecture.

Functional and quality-of-service requirements are not independent. Changing QoS requirements affects the functionality that can be delivered just as changing the functionality may require updated QoS requirements. These must be balanced in the context of overall system functionality and scope.

Another critical element of system design is the need to balance the overall system scope with individual project scope. Projects are typically time-bound and resource constrained while technical architectures and programs are long-lived and must address requirements that evolve over time. Achieving this balance requires organizational diligence.

3. COMMUNICATION QUALITY OF SERVICE REQUIREMENTS

3.1. Service Provider Requirements

Service providers have a wide range of requirements arising from the need to maintain a stable and reliable grid under all circumstances. FERC has defined a set of 14 DR program categories that range from long-term energy markets down to 4 seconds regulation. [2]

A service provider needs to evaluate the communication requirements for all programs that will be implemented in order to avoid the implementation of a fragmented system that will be both expensive and brittle as it tries to evolve to accommodate higher levels of security and performance.

3.2. Customer Requirements

Customers typically have a wide range of energy consuming and producing resources with a variety of capabilities based on the different criticalities of the resources, their device communication speeds, and energy consumption. An effective communication architecture and design must accommodate the inclusion of all devices to effectively participate in the DR programs for which their characteristics are matched.

3.3. Stability and Reliability Requirements

The engineering basis for achieving closed-loop system stability lies in the field of automatic control theory and the application of negative feedback in such a way as to counteract the effects of system perturbations. This requires that loop gain and phase shift be constrained based on the dynamics of the system. Inadequate control system performance can adversely affect stability through changes in loop gain and phase shift.

Demand response systems are wide-area feedback systems that are subject to instability if the system and communication response characteristics (i.e. latency) are not properly matched with the required demand response performance. High performance, reliable, low-latency communication architectures minimize loop phase shift and promote DR system stability.

3.4. Supportability and Maintainability Requirements

As demand response grows and requirements change over time, the capability of a system to support new requirements without major investment significantly increases system flexibility and cost-effectiveness by minimizing program life-cycle costs and time-to-market.

3.5. Security Requirements

Security is critical for demand response applications and arises from the large and diverse customer loads. Large industrial loads can impact the grid directly as can the aggregation of smaller loads. Demand response security requirements have been addressed by UCAIug [3].

3.6. Scalability and Availability Requirements

Demand response networks are large, wide-area networks that can incorporate very large numbers of diverse customer nodes and devices. The communications system must be architected to scale predictably from thousands to millions of end points and not degrade as the number of these end nodes increases.

To aid scalability, a hierarchical communication system is needed to allow for federation into many communication domains. Communications servers must therefore allow for the federation of communications responsibility.

3.7. Performance Requirements

Performance includes both system and communication; 1) latency (i.e. time delays) and through-put (i.e. messages per second).

Achieving high-performance demand response communications requires low-latency messaging at scale.

Communications latency and through-put are dependent upon all the lower layers of the GWAC [4] [5] stack operating efficiently including the physical network, payload encoding, messaging patterns, and message processing.

In the case of Internet broadband and wireless communications based on TCP/IP sockets, the key factors affecting performance include payload size, messaging patterns and message processing. Payload size is determined by message content and encoding rules. Small XML and binary-encoded payloads provide the highest performance.

4. TECHNICAL COMMUNICATION REQUIREMENTS

The following are technical communication requirements for the management and control of devices that communicate over the Internet that support the QoS requirements above.

Note: The use of the term "controller" in this context implies that it has the responsibility for interacting with many devices in hierarchical roles and relationships. This is the same as "Virtual Top Node" and "Virtual End Node" as defined by OASIS in Energy Interoperations.[2] Devices and controllers are collaborators (i.e. endpoints) within the communication infrastructure. Other terms may be used for other applications, as appropriate.

These are general requirements and are not specific to any given application. They should be evaluated on a system architectural basis for all demand response applications.

4.1. Identity

1. Devices shall have a unique identity across the demand response wide-area network. This identity may be used as a logical address for communications. A fixed IP address cannot be assumed as a unique identity. This is required so that DHCP can be used and nomadic devices can be addressed.

2. It must be possible for a controller or device to address a message to a specific endpoint, where the destination may be a device or a controller. This is a basic communication pattern that is required. However, depending upon the application, it may be desirable to either enable or restrict peer-to-peer communications between devices.

3. Group communications shall be supported, where a controller can address a message to all devices in a group. Devices may have membership in zero or more groups. This means that each message shall have a single source, but potentially many destination addresses where a destination address may be a group address that is maintained and managed by the communication infrastructure. This is a specification of a pub/sub mechanism, as might be used for issuing pricing signals for some applications. It's a basic communication pattern that is required.

4.2. Security

4. Devices shall be configured with credentials that enable them to make authenticated and authorized connections to a trusted communication infrastructure. A device must connect to the communication infrastructure, and only authenticated connections are allowed. Communication over the infrastructure is restricted to a trusted set of participants (e.g. controllers and devices). This supports the fact that it is easier for a device to find the communication infrastructure than it is for the infrastructure to find a device in a high-scale system.

5. Devices shall not be required to accept inbound connections and shall not require open ports in firewalls to allow them to communicate over the Internet. Devices will only make outbound connections to the communication infrastructure using their credentials. This recognizes and addresses a significant security barrier for the widespread deployment of devices.

6. Communications over the Internet shall be encrypted.

7. There shall be a mechanism to update security credentials, e.g. withdraw certificates.

8. Devices or controllers shall support a role-based security model which defines access down to parameter or message-type level.

9. Devices or controllers shall be capable of simultaneously maintaining multiple connections to different communication collaborators with different trust levels.

4.3. Performance

10. Devices must be able to receive messages asynchronously, without the need to poll a controller. This is essential for real-time communications.

11. It must be possible for a controller to readily determine the presence and state of a device.

12. The virtual order of messages shall be preserved. In this way a device will see all of the commands issued by a controller in the proper order, and a controller may see all of the events issued by a device in the proper order.

[2] http://docs.oasis-open.org/energyinterop/ei/v1.0/energyinterop-v1.0.html

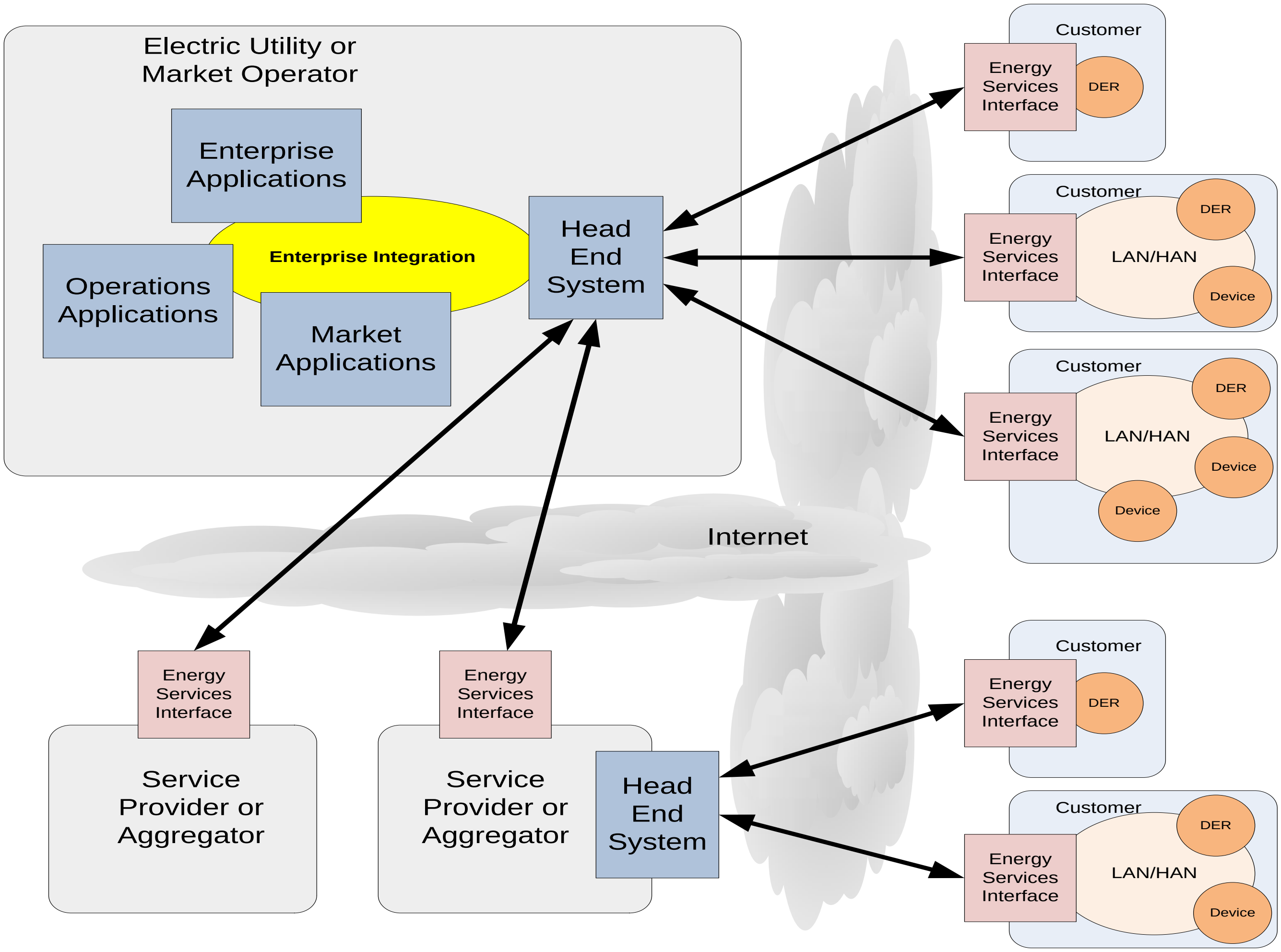

Figure 1 – Typical Demand Response Network

4.4. Economic

13. Communications must be compatible with IETF Internet standards. Protocols should be based upon existing industry standards and must not rely upon proprietary software or licensing for the development of endpoints that use the interfaces.

14. Must establish a low end technology threshold for end point devices and related software components in order to avoid limiting the specification in order to support 'constrained' devices. Low end technology requirements include:

    a. The ability to make a secure internet connection using IETF defined standards [8].

    b. The ability to parse and generate XML messages.

    c. The ability to be externally/locally configured and managed as necessary to define identity, connection parameters and parameters that describe capabilities as needed for participation in DR programs.

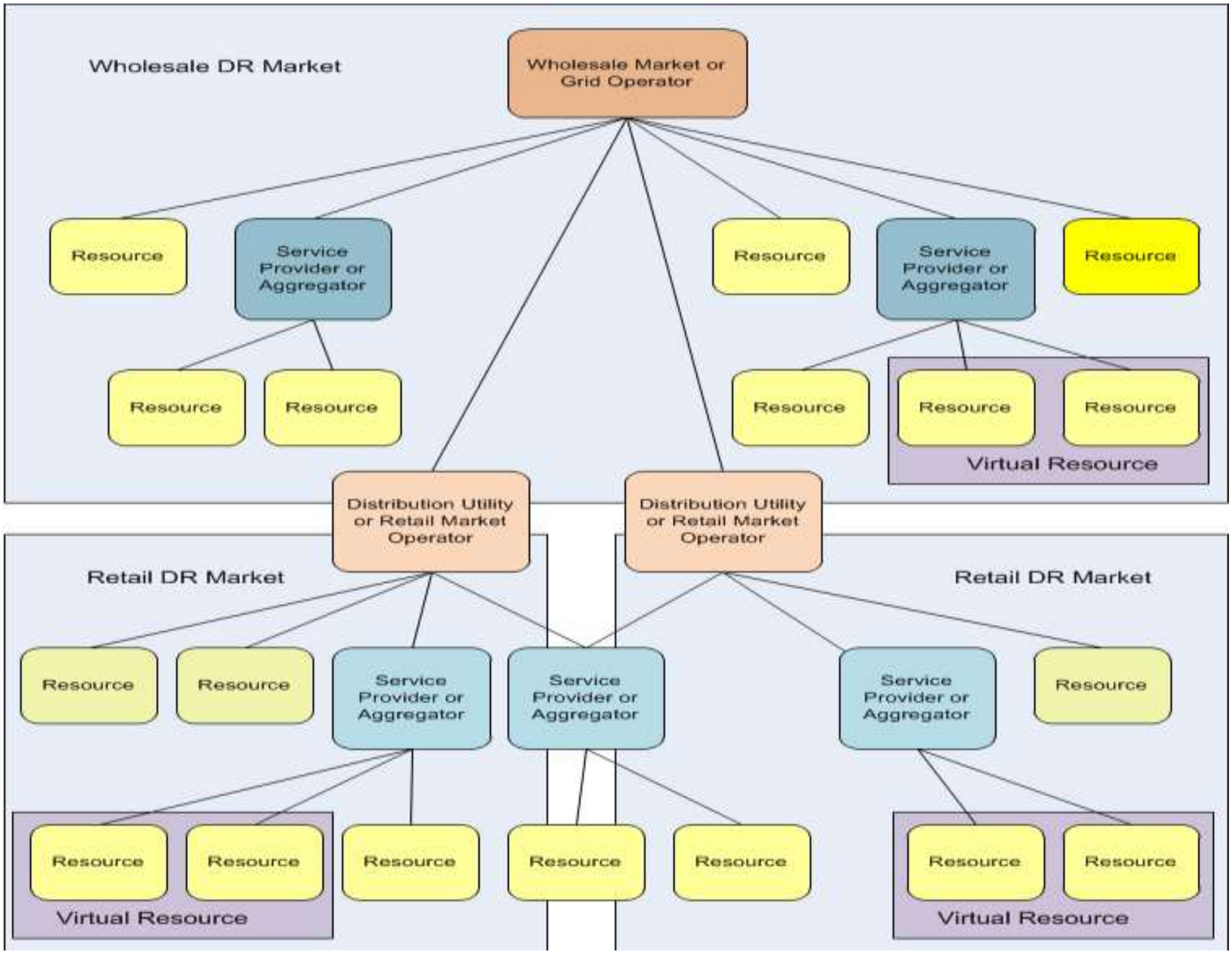
Wholesale DR Market
Wholesale Market or Grid Operator
Resource
Service Provider or Aggregator
Resource
Resource
Resource
Service Provider or Aggregator
Resource
Resource
Resource
Resource
Resource
Virtual Resource
Distribution Utility or Retail Market Operator
Distribution Utility or Retail Market Operator
Retail DR Market
Retail DR Market
Resource
Resource
Service Provider or Aggregator
Service Provider or Aggregator
Service Provider or Aggregator
Resource
Resource
Resource
Resource
Resource
Resource
Resource
Resource
Virtual Resource
Virtual Resource

Figure 2 – Hierarchical Demand Response Topology

15. Communications must be based on open technologies and standards and not be tied to a single programming language or operating system, and conversely should allow for implementations using a variety of commonly used programming languages and operating systems, including those found on mobile devices.
16. Devices must be easy to implement without significant technical barriers, either by direct generation/translation of the wire protocol or aided by API libraries.
17. Communications must be scalable and support thousands of endpoints with a single server, along with allowing for federation using many servers.

5. ARCHITECTURAL CONSIDERATIONS

5.1. Demand Response

A common demand response network structure (Figure 1) consists of a demand response market operator that communicates; 1) directly to resources; local controllers that manage resources at a location or 2) through service providers who may then in turn manage resources through a variety of means. A broader DR topology view (Figure 2) illustrates the hierarchical relationships between wholesale markets, retail markets, resources and virtual resources. These are examples of distributed communications architectures. Ideally there could be one communication network infrastructure that could be leveraged at all levels of this demand response hierarchy, with variations in message payloads as needed within each level.

5.2. Scalability

Scalability can be achieved using either scale-up or scale-out architectures. Scale-up architectures typically refer to monolithic structures that are large enough to accommodate the application. An example of this is the replacement of a small hardware server platform with a large server that has more memory and processors. Scale-out architectures refer to the ability to add more servers to an existing network to meet increased load requirements.[3]Scale-out architectures have proven very valuable in cloud computing infrastructures[4], Big Data NOSQL storage[5], high performance parallel computing applications such as Hadoop map-reduce[6] and everyday applications such as email and instant messaging.

One design approach to scale-out networks uses a logical federation topology (Figure 3) which decomposes the network into domains with the capability of routing messages between domains as required. This can best be illustrated using an example everyone is familiar with; email. An email address is composed of a username and a domain such as "dhardin@enernoc.com" and "sneumann@uisol.com". Messages sent between these two usernames are routed by the system to the two unique domains "enernoc.com" and "uisol.com". The same basic topology is used for texting and instant messaging to achieve very high scale-out H2H (human-to-human) communications with the trend toward achieving real-time communications (Figure 4). The major difference between email federation and near real-time federation is that email is based on multi-hop and real-time on single-hop federation.

Domains can also be used for the creation of larger hierarchical topologies, where a member of one domain (e.g. a wholesale DR market) can be the manager of a lower level domain (e.g. a retail DR market or aggregator). Within a hierarchical topology there can be as many levels as needed. The fact that there are many levels can (and should) be transparent to a higher level domain, from both functional and technical perspectives. An example of this is shown in Figure 2, where there would be minimally three domains (one for the wholesale market and two for retail markets), and potentially five more if each service provider or aggregator implemented their own domain for management of resources.

Distributed networks using hierarchical topologies and scale-out federation have also been used extensively within the industrial automation domain to provide real-time data acquisition and control of large industrial processes using DCS (Distributed Control Systems).[7] In addition to logically subdividing the network based on the characteristics of the industrial processes, these systems also rely upon efficient, change-driven, PUSH communications to achieve high performance and throughput.

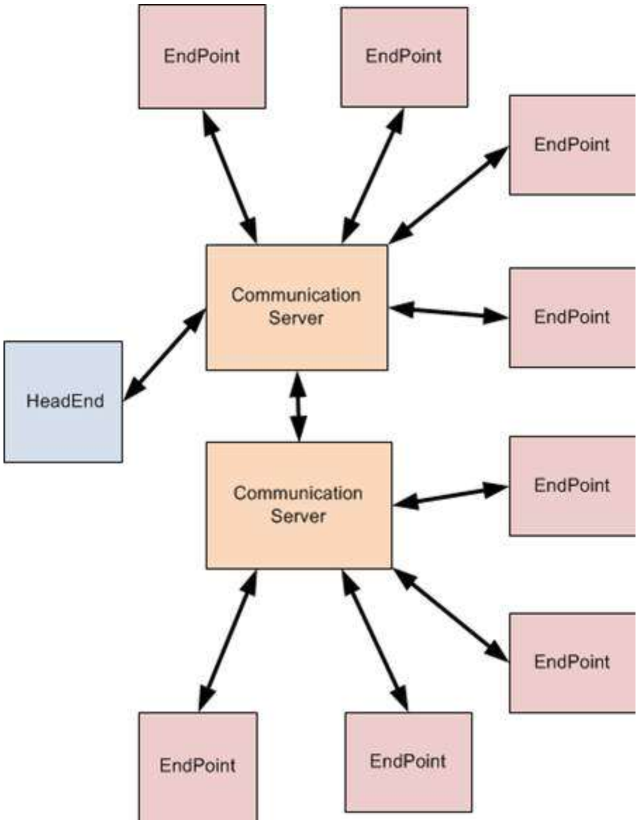

Figure 3 – Logical Federation Topology

[3] http://www.intel.com/content/dam/www/public/us/en/documents/white-papers/cloud-computing-journey-to-cloud-v2i1-paper.pdf

[4] http://www.newvem.com/aws-guide/scalability-management-guide/

[5] http://nosql-database.org/

[6] http://hadoop.apache.org/

[7] http://en.wikipedia.org/wiki/Distributed_control_system

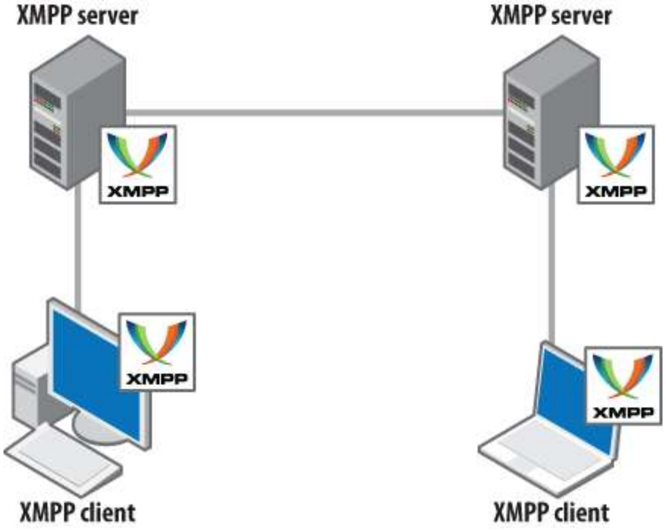

Figure 4 – XMPP Single-Hop Server to Server Federation for Instant Messaging [7]

5.3 Performance

As previously indicated, communication performance relates to the throughput and latency of messages sent through a connection between two systems. Throughput refers to the quantity of messages and is measured in messages per unit of time. Latency refers to the time delay between when a message is sent and when that message is received and is measured in units of time. Both are very important as a system scales but each has benefits in different applications. An example relates to the differences between a typical personal computer (PC) operating system (OS) and a real-time operating system for a SCADA (Supervisory Control and Data Acquisition) controller. A PC OS is designed to manage thread execution in a manner that achieves the highest overall system processing throughput while the real-time OS is designed to achieve the lowest deterministic response latency to events. Many applications, such as large-scale demand response, must maintain both high throughput and low latency as scale increases.

Low latency can be best achieved using a PUSH interaction pattern where the message is sent through a connection immediately when ready. A simple example of a PUSH interaction is a phone that rings when someone wants to talk. This is in contrast to a PULL interaction where a system is polled at some interval to test for the existence of a message that is ready to be received. In the phone example, this is the same as picking up the phone every few minutes to see if someone is ready to talk. Another example of PULL is to keep pressing the refresh button on a web site in hopes that a change occurred. The PUSH communication pattern provides the lowest latency and most efficient communication pattern in terms of minimizing communication traffic. [6]

5.4 Security

Communication security is critical due to cyber threats on both the customer and bulk power system. Large numbers of customer facility energy management systems represent an attack surface that has the potential to disrupt the bulk power system. Invalid signals sent to customers' systems can interrupt and compromise commercial and industrial operations and can result in harm to equipment and personnel. Invalid signals sent from customers to service providers can cause misinformation and result in potentially harmful actions.

An important aspect of logical, multi-domain, multi-tier architectures is that they must be constructed from server and client components that can establish strong trust relationships. These trust relationships require that the following five areas of security be addressed: 1) authentication, 2) authorization, 3) confidentiality, 4) integrity and 5) non-repudiation. Authentication refers to validating the identity of a user or code. Authorization refers to validating the authority of a user or node to perform actions. Confidentiality is the ability to encrypt data in order to prevent its access and integrity is the ability to detect data tampering. Non-repudiation is the ability to ensure that messages are sent and received by those that claim to have sent and received. The digital techniques used to mitigate these security issues include; 1) authentication using digital certificates (i.e. X.509), 2) authorization using digital certificates, 3) confidentiality using message encryption with digital certificates, 4) integrity using message signing with digital certificates and 5) non-repudiation using a combination of the above including message signing using digital signatures, time-stamps, and encryption.

These techniques however are insufficient to adequately protect a system against the wide range of potential cyber threats. Many enterprises have adopted a fundamental policy to manage access to internal systems using firewalls that block incoming connections and enable but limit outgoing connections. Successful applications, such as cloud computing service bus messaging[8] and device instant messaging have adapted to this policy by only requiring outbound connections. Once an outbound connection is established, the digital techniques described above are used to ensure that the connection is secure while in operation.

[8] http://blogs.msdn.com/b/dachou/archive/2011/03/24/internet-service-bus-and-windows-azure-appfabric.aspx

NISTIR (NIST Internal Report) 7628[9] provides further guidance on cyber security.

6. SUMMARY

Demand response program performance requirements range from day-ahead to fast ancillary and regulation services. Low-performance communication architectures can satisfy the needs of some customer resources but high performance architectures can satisfy the needs of all program participants.

The development of communication architecture for demand response that can scale while enabling customers to participate in a range of programs requires a secure, low-latency communications architecture that delivers messages when needed with efficient message encoding while providing strong security and not requiring open firewalls. These basic requirements have already been addressed using existing architectures and technologies that have been field-proven for near-realtime applications at very large scale. An example open technology that supports this architectural model is the IETF Extensible Messaging and Presence Protocol (XMPP) [8] [9].

7. RECOMMENDATIONS

7.1. Recommendations for Research

1. Closed-loop modeling of large-scale demand response systems to aid in quantitatively identifying system performance specifications and constraints.

7.2. Recommendations for Standards

1. Launch an effort within an appropriate standards coordination body such as the Smart Grid Interoperability Panel (SGIP) or IETF to define the requirements for a next-generation real-time Internet communications protocol standard that leverages IPV6 to provide; 1) congestion-resistance, 2) very low latency, 3) high scalability, 4) connection rerouting and 5) connection failover that would support real-time messaging for a wide range of critical cloud-based Internet applications.

Biographies

Dave Hardin, EnerNOC

Dave is Sr. Director of Smart Grid Standards at EnerNOC. He is active in a number of Smart Grid initiatives including the Smart Grid Interoperability Panel as vice-chair of the Architecture Committee and chair of the Industrial-to-Grid Domain Expert Working Group, the OpenADR Alliance Board of Directors, and the OPC Foundation Technical Advisory Council and is a member emeritus of the U.S Department of Energy's GridWise Architecture Council.

Scott Neumann, UISOL

Scott is Chief Technology Officer at UISOL. In that role he leads architecture and integration efforts for electric utilities, grid operators and market operators, but has also designed and developed a variety of commercial products that are used in the electric utility market space. He also serves as the US Technical Advisor for IEC TC57. He has been active in standards development efforts and has been the project leader for the IEC 61968-9 and IEC 61968-100 standards efforts and is currently the architecture lead for IEC 62746. Scott is a senior member of the IEEE.

REFERENCES

[1] *SGIP B2B I2G Energy Services Interface Whitepaper*, 2008, http://www.gridwiseac.org/pdfs/interopframework_v1_1.pdf

[2] *National Assessment & Action Plan on Demand Response*, U.S. Federal Energy Regulatory Commision, http://www.ferc.gov/industries/electric/indus-act/demand-response/dr-potential.asp

[3] *Security Profile for OpenADR*, 2012, The UCAIug OpenADR Task Force and SG Security Joint Task Force

[4] *NIST Framework and Roadmap for Smart Grid Interoperability Interoperability Standards Release 1.0*, 2009, Office of the National Coordinator for Smart Grid Interoperability, National Institute of Standards and Technology, U.S. Department of Commerce, www.nist.gov/public_affairs/releases/smartgrid_interoperability.pdf

[5] *GridWise Interoperability Context-Setting Framework V1.1*, 2008, GridWise Architecture Council, http://www.gridwiseac.org/pdfs/interopframework_v1_1.pdf

[6] *Push vs. Pull: Implications of Protocol Design on Controlling Unwanted Traffic*, Zhenhai Duan, Kartik Gopalan, Yingfei Dong, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.125.9223&rep=rep1&type=pdf

[7] *XMPP: The Definitive Guide Building Real-Time Applications with Jabber Technologies*, Peter Saint-Andre, Kevin Smith, and Remko Tronçon, 2009, O'reilly Media, ISBN: 978-0-596-52126-4,

[9] http://csrc.nist.gov/publications/PubsNISTIRs.html

http://oriolrius.cat/blog/wp-content/uploads/2009/10/Oreilly.XMPP.The.Definitive.Guide.May.2009.pdf

[8] *IETF RFC 6272, Internet Protocols for the Smart Grid,* 2011, the Internet Engineering Task Force

[9] *IETF RFC 6210, Extensible Message and Presence Protocol,* 2011, the Internet Engineering Task Force