ENCOURAGE

Embedded iNtelligent COntrols for bUildings with Renewable generAtion and storaGE

Grant Agreement No.: 269354

D4.1 – ENCOURAGE Communication architecture for device interoperability

David Jorquera, Maarten Los, Abdel Rahman, Luis Lino Ferreira, Rodrigo Ferreira

Document Number	D4.1
Document Title	ENCOURAGE Communication architecture for device interoperability
Version	1.0
Status	Final
Work Package	WP4
Deliverable Type	Report
Contractual Date of Delivery	M18
Actual Date of Delivery	M18 - 3/12/2012
Responsible Unit	ATOS
Contributors	ATOS, ISEP, SLX, ISA
Keyword List	SECURITY, PROTOCOL, CLOUD, RABBITMQ
Dissemination level	PU

Amendment History

Version	Date	Author (Unit)	Description
0.1	14/09/2012	ATOS	First draft
0.2	26/11/2012	ENORD/EZMON	Review
0.3	29/11/2012	ISA	Update
0.4	3/12/2012	ATOS	Incorporate review comments
1.0	3/12/2012	AAU	Quality control/Formatting

Executive Summary

This task is focused on ensuring that all devices, systems, services, protocols and agents are fully interoperable and that security & privacy of communications are maintained. In this sense the task is responsible for ensuring confidentiality, integrity and availability of data in communications between the ENCOURAGE devices. The components will also provide solutions to protect users' privacy. It is important that no information can be attained by unauthorized parties.

The aim is to protect users' information from misuse that might arise during a communication of this information over an insecure channel. This can achieved, for example, by providing a component that runs authentication protocols and that decides upon a session key before starting the communication of sensitive data. The possibility of defining and managing privacy policy that may guarantee a correct usage of the personal data is also considered here. Moreover, the problem of protecting the integrity of data of the sensors from attempt of tampering made by the users themselves is also addressed by this task.

Glossary of Acronyms

ACRONYMS	DEFINITION
SLA	Service Level Agreement
NAS	Network Attached Storage
SAN	Storage Area Network
SPoF	Single Point of Failure
SSO	Single Sign On
JAAS	Java Authorization and Authentication Service
SOAP	Simple Object Access Protocol
HTTP	Hypertext Transfer Protocol
LDAP	Lightweight Directory Access Protocol
EJB	Enterprise Java Beans
JSP	Java Server Pages
SAML	Security Assertion Mark-up Language
BC	Business Case
MI	Messaging INnfrastructure
CD	Configuration and Diagnosis
UI	User Interface
LM	Load Management Module
DSEB	Decision Support Module
FC	Forecasting Module
HGW	HAN Gateway
DB	Database
DEV	Devices
VD	Virtual Device
CEP	Complex Even Processor
MPG	Middleware Plug-in Gateway
BI	Business Intelligence
DMZ	Demilitarized zone
VPN	Virtual Private Network
SLA	Service Level Agreement
NAS	Network Attached Storage
SAN	Storage Area Network
SPoF	Single Point of Failure
SSO	Single Sign On
JAAS	Java Authorization and Authentication Service
SOAP	Simple Object Access Protocol
HTTP	Hypertext Transfer Protocol
LDAP	Lightweight Directory Access Protocol
EJB	Enterprise Java Beans
JSP	Java Server Pages
SAML	Security Assertion Mark-up Language
SC	Supervisory Control
EB&BI	Energy Brokerage and Business Intelligence Block

Table of Contents

Table of Figures

Table of Tables

1. Introduction

This document constitutes deliverable D4.1 addressing Communication architecture technologies and protocols, it is part of WP4 which is responsible for providing infrastructures and models to access and integrate sensor/device information from heterogeneous sources.

The development of the Cloud Infrastructure is also part of this of this WP, as well as the messaging infrastructure.

Security methods have been analysed and described in order to provide a clear set of technologies that will help the project success.

Chapter 2 gives an introductory overview of all actors, technologies and scenarios defined in the project.

Chapter 3 gives an overview of the security mechanisms that can be used to protect the communications between ENCOURAGE HAN Gateways and the ENCOURAGE Middleware Plugins.

Chapter 4 is based in security issues related to the Cloud Infrastructure and Chapter 5 overlook the security aspects of the messaging broker.

It takes as input the efforts of WP2 and is an input for the other tasks of WP4.

2. ENCOURAGE Architecture

Taking as input the high level depiction of encourage scenarios the big picture ought to be introduce as an introductory point of all actors, technologies and scenarios.

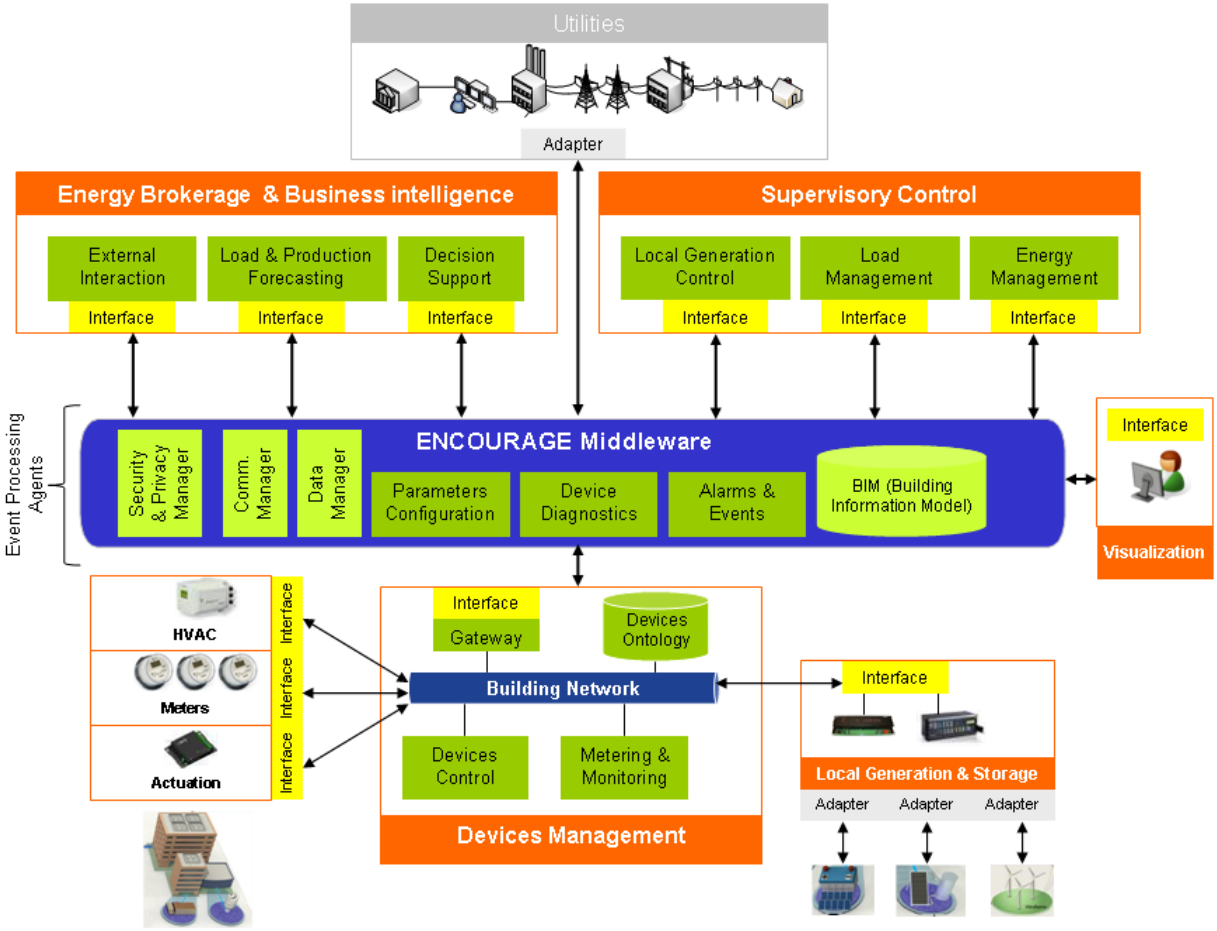

Figure 1: Encourage Architecture

The figure above represents the ENCOURAGE architecture that is divided into four logical blocks:

- Device Management
- Middleware
- Supervisory Control
- Energy Brokerage and Business Intelligence

The WP4 is responsible for the development of the Middleware. And the WP 4.1 will provide the Cloud services required for the deployment of the middleware services.

The main functionalities of the middleware services are to provide a uniform data communication and processing bus between the ENCOURAGE platform elements.

The Middleware is constituted by:

- Messaging Infrastructure
- Database
- Configuration and Diagnostic module
- Complex Event Processor.

The "Messaging Infrastructure" will link all the higher ENCOURAGE modules, EB&BI and SC between them, connects those applications with HAN devices and with the other modules of the Middleware layer. This solution allows reducing the complexity of inter-application communications since it works as a message bus, thus applications use only one communication protocol and are only required to maintain one connection with the message bus.

The functionalities required by the messaging infrastructure are:

- Publish/subscribe communications
    - Configure events and notifications on the Messaging Infrastructure. It should include one interface for basic configuration (a simple subscription of an event) and an advanced interface for:
        - Send events;
        - Receive notification of events;
- Client/Server communications
    - Send/receive messages

The Database component will be used to permanently store ENCOURAGE system information, it may contain historical data for every variable/device subscribed, which can after be used for consumption prediction.

The Database is responsible for storing data both historical and current within the ENCOURAGE platform. This data can be retrieved by calls made to the Middleware for the purposes of historical analysis, and consumption or production prediction.

Functionalities required:

- Store data
- Access the virtual representation of a device

- Convert between the Middleware representation and the Virtual Device representation
- Read/write access to virtual devices representation

The Complex Event Processor (CEP) is an engine that is capable of detecting asynchronously, independent incoming events of different types and generating a complex (synthesized) event out of these events. The key difference with other concepts such as Business Rules and Event Stream Processing is that the incoming events can be asynchronous and of different types. Another difference is that the CEP has temporal awareness.

Functionalities required for CEP:

- Configure a complex event: this functionality also requires the interaction of this module with the Messaging Infrastructure to configure to which event it subscribes;
- Start, stop commands to activate, deactivate certain rules.

The module "Configuration and Diagnosis Module" will manage the configuration of the Middleware, HAN and Middleware Gateways and Devices, whose configuration data will be stored in the database. At system start-up this module is responsible for the initialization of all ENCOURAGE modules as well as for the setting of specific parameters on the HAN Gateways.

3. Communication Security

This Section gives an overview of the security mechanisms that can be used to protect the communications between ENCOURAGE HAN Gateways and the ENCOURAGE Middleware Plugins. The objective is to provide confidentiality, data integrity, authentication and availability. This interconnection will be based on a Restful interface and it is assumed that all communication is routed through public networks, which belong to different Internet Service Providers (ISPs).

We start by identifying the existing threats to this interconnection, afterwards we discuss the possible counter-measures and we end the section by proposing the solution that will be implemented in ENCOURAGE.

3.1 Threads for HAN-to-Middleware communications

Security in smart grid applications must be carefully implemented to maintain the reliability and usability of the system. Smart Grids have a number of potentially significant security requirements that must be addressed, which include:

- Confidentiality: Message content should be accessed by authorized users only. Achieved by using encryption.
- Integrity: Making sure that messages are not altered (in transit, or later) without detection. Achieved by using hashing technologies.
- Authentication: sender and receiver need to confirm the identity of each other. Achieved by using digital certificates or username/password.
- Availability: services must be accessible and available to authorized users. Achieved by using Audit logging.

The known security attacks to which smart grid HAN Gateway to Middleware plugins systems may be susceptible can be classified in several types: denial of service, passive eavesdropping, replay attack, address impersonation and session hijacking.

Denial Of service

In this type of attacks the attacker sends a huge number of requests to the server to overload it. The server will not be accessible to the legitimate users, so this makes the server no longer functional. This kind of attack can have a single machine as its source, or it can be originated from multiple machines. This last type is usually called Distributed Denial of Service (DDoS). In Smart Grids, Denial of Service attacks try to delay, block or corrupt data transmission in order to make network resources unavailable to devices that need the exchanged data. For example, an attacker can physically connect to a communication channel and generate legitimate traffic over the channel.

Afterwards, a high volume of traffic can be used to delay data transmission and to overload control devices. In order to protect the network from this kind of attacks, strong authentication is required on all data exchanges.

Passive Eavesdropping

Passive Eavesdropping is type of a theft information attack. A passive eavesdropping attack happens when an attacker starts to listen to the traffic that is transferred between two endpoints.

The attacker in passive eavesdropping needs to have access to the traffic, this can happen in different ways. An attacker can get access to a network and connect a host to the network. Sometimes a thief is able to receive packets transmitted by radio signals if he is close enough to the wireless network. The best solution would be to use the end–to-end encryption method on all traffic, which makes eavesdropping attacks useless.

Replay Attack

Replay attack is an attack in which a valid data transmission is repeated maliciously. The logic of the attack is based on sending messages that reserve remote resources or makes the remote system perform an action, to control the remote system. In most cases, the attacker does not have the knowledge of the semantics of the request message, hence he is limited to repeating what he considers a valid request message. Authentication alone can prevent the denial of service attack, but it cannot protect from a replay attack, because the attacker can have a copy of the valid request message, buffer it, and then resend it later.

One approach to prevent this kind of attack involves associating each message to a different identification field, and including the identification field in the message. This approach force the request message to change for each subsequent communication, and the attacker would need to be able to understand the semantics of the message, and change it properly to create a new valid request message. Moreover, time stamping can be used to estimate when the identification field was generated, to ignore old messages.

Address Impersonation

Impersonation attacks happen when the attacker is able to change its packets' source address to the address trusted by the attacked node.

An example of address impersonation in an encryption-enabled environment, is the man-in-the-middle (MitM) attack. Let us suppose that A wants to communicate with B. A will perform a two or three way handshake with B, then exchange cryptographic keys and start communicating with privacy. Let us suppose that C wants to intercept the communication. It can tell to A that it is B. When A starts the handshake with B, C can receive the authentication data of A and use it to perform handshaking with B. The result is that A and B will think to have a direct link with each

other, while they have it with C, which encrypt/decrypt every message in transit with the keys agreed upon with A and B.

Session Hijacking

In session hijacking, the attacker gains unauthorized access to a session between two nodes and intercepts the packets exchanged between. The attacker impersonates one of the nodes, alters or discards the original packets of the session, and finally the attacker takes over the whole session. The protection against session hijacking is the same as passive eavesdropping by providing end-to-end encryption with authentication.

3.2 Security Solutions

Various technologies have been developed to defend networks from security threats. These technologies usually provide confidentiality, integrity and authentication to communications. The classic technologies are cryptography, key management, authentication, auditing and firewalling. These technologies are considered the basic building blocks for current security solutions. Figure 1 illustrates Security Requirements, Threats, Attacks, and Solutions.

Encryption methods are either symmetric (secret key) encryption or asymmetric (public key) encryption. In symmetric algorithms the same key is used for both encryption and decryption of the message, whereas asymmetric algorithms use different keys for encryption and decryption, the private key for encryption and the public key for decryption. The private key will be kept secret and never shared. The public key is created from the private key and is exchanged over insecure communication channels. After that, each user combines the public key of the other user with its own private key and calculates the same shared secret number. The shared secret number will be converted into a shared secret key. This shared secret key should never be exchanged over an insecure communication. An example of asymmetric cryptographic algorithm is RSA [2].

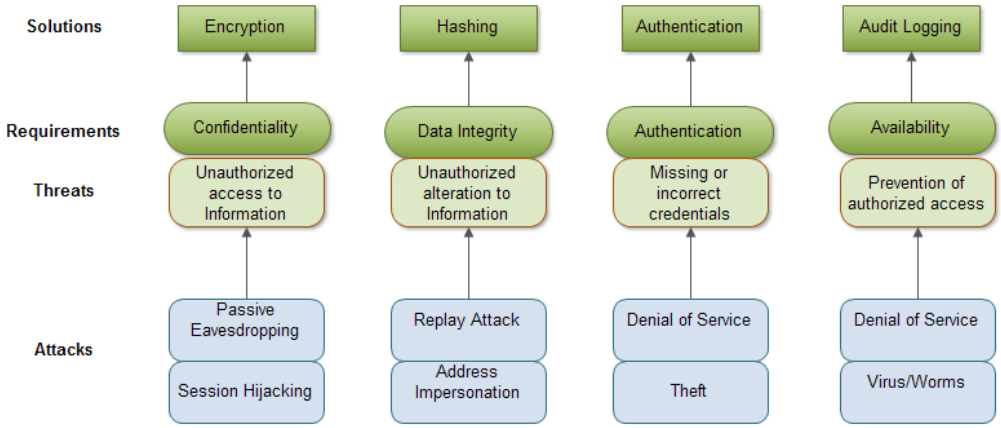

Figure 2 Security Requirements, Threats, Attacks, and Solutions

Symmetric-key algorithms can be divided into block ciphers and stream ciphers. Block ciphers take a fixed number of bits and encrypt them as a single unit, while stream ciphers encrypt the bits of the message as they are received. Data Encryption Standard (DES), Triple-DES and Advanced Encryption Standard (AES) are examples of symmetric block ciphers algorithms [1].

DES is based on a symmetric key algorithm with a 56-bit key. Currently, DES is considered to be insecure for many applications due to the small number of bits used by the key. The algorithm is believed to be secure in form with Triple DES (3DES), although there are theoretical attacks that can break 3DES. In recent years, the cipher has been superseded by the AES. Triple Data Encryption Algorithm (3 DES) is a block cipher based on the repetition of the Data Encryption Standard (DES) for three times. 3DES has a key length of 168 bits (three 56-bit DES keys).

The Advanced Encryption Standard (AES) is a block cipher algorithm adopted by the National Institute of Standards and Technology (NIST) to replace DES. AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits. AES is relatively simple to implement, requires little memory and offers a good level of security which overall preferred to the other proposed algorithms.

Authentication is the process of making sure that the message is coming from its intended source and going to its intended destination. Authentication is commonly achieved by the use of username/password, pre-shared keys and digital certificates.

When using pre-shared keys, a shared secret key is previously exchanged between two endpoints using a secure communications channel. Otherwise, when using a digital certificate then there is a digital signed document that validates each endpoint. The signature is obtained from a specific certificate authority.

A firewall is either hardware or software used to enforce access control policy between networks. The firewall filters the incoming packets, where it rejects any unauthorized packets. Another type provides proxy services, data verification and authenticates service requests.

Auditing is a mechanism used to log system activities. It has become an important technology in network security. Intrusion detection system (IDS) is one of these technologies, IDS is software or hardware device passively listens to the network traffic and when the IDS detect malicious traffic, it sends an alert to the management station.

Different types of communication media can be used to connect HAN Gateways with the ENCOURAGE Middleware, which can be a combination of different technologies like Ethernet, ADSL, Fibre Optics, GSM, UMTS, T1, SONET, just to mention a few. For such systems security can be applied at different layers of the Open Systems Interconnection model (OSI model): Application, Transport, Data Link and Physical Layer. Due to the different types of media to be transverse by ENCOURGE packets, the only viable solution is to use Application, Transport or Network layer security capabilities. Figure 2 shows Multi-layer security protocols.

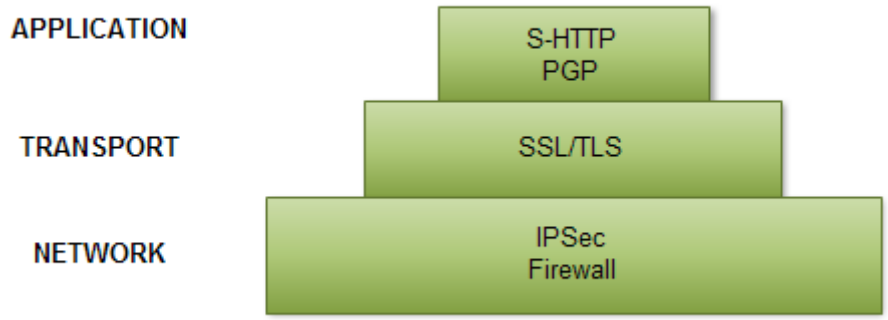

Figure 3: Multi-layer security

3.2.1 Application Layer

The Application Layer allows applications to use network services. The protocols and programs that operate in the Application Layer include: FTP, Telnet, Remote Desktop, Web Browsers, HTTP, Email Clients and more. Each one of these programs uses its own protocol making them more prune to security vulnerabilities. Two possible solutions are to use Pretty Good Privacy (PGP) or Secure HTTP (S-HTTP) [3].

PGP is usually implemented as software for encryption and decryption, which ensures the privacy and authentication of data communications. PGP uses existing cryptographic algorithms such as RSA, IDEA or MD5. PGP also supports digital signatures, key management, secrecy, and data compression.

S-HTTP is an extension of HTTP that allows the secure exchange of information between an HTTP client and an HTTP server using encryption. S-HTTP is different than HTTPS, since the last is basically HTTP on top of SSL/TLS for secured transactions. S-HTTP encrypts individual messages, while HTTPS encrypts the communication channel. Thus, S-HTTPS cannot be used for VPN security, but HTTPS can.

3.2.2 Transport Layer

Transport layer provides end-to-end communication services for applications. The most commonly used protocols are Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP).

Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL) are encryption protocols for secure data transmission over the Internet [4]. Since version 3.0, the SSL protocol is being developed under the name TLS. SSL/TLS is implemented on top of the Transport Layer protocols and typically runs on top of TCP for reliable data transfer. SSL/TLS use asymmetric

cryptography for the exchange of a symmetric key, which is then used for symmetric encryption for privacy, authenticity, and to preserve the integrity of messages. This protocol is widely used in applications that use the Internet, such as Web browsers, email access, instant messaging, and IP-Telephony (VoIP).

The SSL/TLS protocol consists of two sub-protocols: SSL Handshake Protocol and SSL Record Protocol. SSL negotiates a state full connection by using a handshaking procedure between clients and servers. The main steps of the protocol are:

1. A client connects to a server that supports TLS, and requests a secure connection.
2. The client provides a list of supported ciphers and hash functions.
3. The server selects method from a list provided by the client.
4. The server sends its digital certificate to authenticate. Usually, a digital certificate contains the server name, identity certification and public key server.
5. The client connects to the server using the trusted certificate and confirms the authenticity of this certificate before starting data exchange.
6. The client and server then use the random seed numbers and its public key to compute a common secret, called the "master secret".

This ends the handshaking procedure. A secure connection between client and server is established; the data is encrypted and decrypted using the encryption key.

The SSL Record Protocol is used to secure the connection. It is based directly on the transport layer and provides two different services that can be used individually or together:

- End-to-end encryption using symmetric algorithms. The encryption type is negotiated in advance using the SSL Handshake Protocol and can only be used once for each connection. SSL support symmetric encryption (for example DES, Triple DES and AES).
- Securing the message integrity and authenticity is achieved by a message authentication code, usually HMAC

3.2.3 Network layer

Network layer is responsible for routing the packets through the network. Network layer security provides end-to-end security across a routed network and can provide encryption services, authentication, and data integrity. Once the network security is applied between endpoints, IP traffic flowing between those endpoints is protected. Internet Protocol Security (IPSec) can provide security at the network layer.

IPSec contains three main protocols: the Encapsulating Security Payload (ESP) Protocol, Authentication Header (AH) Protocol and the Internet Key Exchange (IKE) [5, 6 and 7], these

provide confidentiality, data origin Authentication and connectionless Integrity between two end points.

The Authentication Header (AH) is used to provide integrity and data origin authentication, and protection against replays but it does not provide confidentiality protection. Encapsulating Security Payload is designed to provide a combination of security services integrity, confidentiality, and authentication of data for both IPv4 and IPv6. ESP may be applied alone, or with the IP Authentication Header (AH). Internet Key Exchange (IKE) is a key management protocol that provides a security association to handle the negotiation of authentication algorithms, encryption algorithms, keys to use and the key's lifetime.

3.2.4 Comparison

The solution chosen for ENCOURAGE is based in secured point-to-point sessions, using SSL/TLS.

For SEP 2.0 implementation, security will follow the specifications of the protocol, with the securing transactions between the HAN gateways and the ENCOURAGE platform servers being guaranteed by using HTTP over TLS [RFC 2818], also known as HTTPS.

For EACS implementation, a VPN connection will be required, offering a complete security solution for communication confidentiality, with minor interference with the business logic of the application. With this approach there is no need to develop specific libraries to ENCOURAGE software if an integration of third-party HAN gateways is required.

Since the Application Layer solution has been rolled opted out it and VPN has been chosen for inclusion in the architecture, there are two possible VPN solutions to be considered. One solution is based on supporting a VPN using IPSec Network Layer Protocol. The second is to consider or to support a VPN over TLS/SLL, but IPSec solution had also to be abandoned since it requires special routers and software, which might not be available. In fact, it was assumed that ENCOURAGE must interoperates with custom HAN Gateways over a complete open network, and building support for SSL/TLS VPN over a HAN Gateway is much proper when considering low-scale hardware in users‘ houses. In the next sections we will elaborate more on this justification.

3.3 Virtual Private Networking (VPN)

A VPN is a private telecommunication network built over an existing public network infrastructure to establish secure end-to-end connection by using encryption techniques and tunnelling [8].

Encapsulation is one of the components the confidentiality of a VPN is built on, with encryption being the other. Encapsulation is often referred to as tunnelling, which is the process of placing an entire packet within another packet when sending it. Private network data is encapsulated through the VPN tunnel and carried over the public network (for example the Internet) so that the tunnelled data is not accessible to anyone examining the transmitted data frames.

VPN can appear in different topologies:

- Site-to-site VPN, linking two or more site-to-site VPN gateways to an internal network over a shared infrastructure using dedicated connections. They are intended to handle secure communications between a company's internal departments and its remote offices, or other similar scenarios.
- Remote Access VPN, allowing remote users to establish secured connections over a shared infrastructure. Remote access VPNs use only a single VPN gateway, where users negotiate a secure connection with a VPN Gateway using a VPN client software.

In the context of ENCOURAGE HAN Gateway to Middleware connection, the Remote Access VPN is the solution that will be implemented.

3.3.1 IPSec vs. SSL VPN

Both IPSec and SSL use standards-based encryption and authentication techniques that secure the private communications over the Internet.

A typical deployment of IPSec VPNs consists of one or more VPN gateways, and VPN client software that must be installed on each remote access user's device. The VPN client is configured to define which packets it should encrypt and with which gateway it should build the VPN tunnel. IPSec clients may need manual configuration making it difficult to use.

IPSec is a network layer VPN technology, which secure all data between two end points, including all applications. IPSec and SSL use different encryption algorithms [12]. IPSec encryption is provided using encryption algorithms including DES, 3DES, and AES. IPSec and SSL use the same authentication technologies such as username and password, X.509 digital certificates, and username and token pin.

Although IPSec is considered to be more secure than SSL, IPSec VPNs are more difficult to deploy and manage, since IPSec needs special-purpose VPN client software and hardware. Moreover, IPSec requires negotiating several security features to create an end-to-end VPN tunnel.

Nevertheless, it is possible to identify several advantages for IPSec: i) IPSec provides security on the network layer and secures everything on top of it; ii) IPSec is an end-to-end security standard for quite long time and has been proven to be a secure and trusted method of securing information; iii) IPSec supports double encrypted tunnel.

But IPSec also presents the following set of disadvantages: i) IPSec is difficult to implement and requires special support in routers; ii) Connectivity can be negatively affected by firewalls or other devices between the gateway and client (NAT devices); iii) Interoperability between IPSec clients to another provider's IPSec servers is usually difficult; iv) IPSec requires client software to be installed in the remote devices, which not all operating systems may support.

The other solution is to use SSL, which is a secure transport protocol used to guarantee the confidentiality and security of transactions in may applications such as e-commerce and online banking. SSL VPN also requires specific client software to be installed in each remote device.

A solution based on SSL/TLS has the following advantages: i) SSL/TLS operates transparently across proxies, NATs, and most firewalls; ii) SSL/TLS is transparent to applications.

Such a solution also has the following set of disadvantages: i) only TCP services are directly supported (the support of UDP is only indirectly achieved by some VPN software packages); ii) SSL/TLS requires more processing resources from the HAN Gateway than IPSec; iii) SSL/TLS tunnel mode is more computational expensive if the implementation needs an external certification authority to sign the digital certificates.

Table 1 summarizes the differences between SSL and IPSec-based VPNs.

	IPSec	**SSL\TLS**
Applications	All applications	All application
Security	High	Medium
Software Required	Client software	Client software
Accessibility	Firewalls and network address translation often interfere with access	All traffic is sent over port 443, which is open through web proxies
Type of connection	Fixed connection	Transient connection
Type of device	Managed corporate device	Varying devices
Protocols	All IP types and services are supported	Only supports TCP services over SSL
Application area	Site-to-site VPNs, secure employee access	Sharing Web, client/server, suppliers and customers

Table 1 IPSec VPNs vs. SSL

SSL/TLS VPN solutions are the most suitable for ENCOURAGE communication security because they are easy to deploy to a broad range of end users. SSL VPNs are a cost-effective choice since they do not require special configuration in routers, and their packets can also easily traverse firewalls and NAT devices without any network topology changes. There are several SSL/TLS VPNs solutions that are able to provide security for ENCOURAGE communication, such as Barracuda, FirePass SSL VPN, University of Tsukuba Virtual Private Network (UT-VPN), and OpenVPN. These solutions are analysed in the next section.

3.3.2 SSL/TLS VPNs Solutions

Barracuda SSL VPN

The Barracuda SSL VPN is an integrated hardware and software solution providing secure, clientless remote access to internal network resources from any computer [13]. The Barracuda SSL VPN provides a control over file systems and Web-based applications. It is designed for providing full network access to the users; the Barracuda is using a secure IP tunnelling client installed on a user's devices. When communication is started, a full IP connection is created to the Barracuda SSL VPN appliance, enabling secure data transformation and allowing the use of any TCP application, like client/server applications. Barracuda's client works on Linux, Macintosh, and Windows operating systems. The Barracuda SSL VPN acts as a web proxy for most intranet websites. The Barracuda SSL VPN can be configured to accept PPTP and L2TP/IPSec connections.

Figure 4 shows the standard deployment of Barracuda SSL VPN.

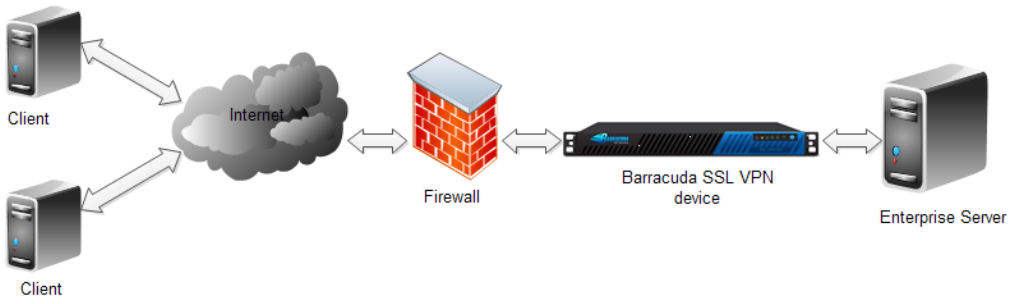

Figure 4: Barracuda SSL VPN standard deployment

FirePass SSL VPN

FirePass SSL VPN is an appliance that provides a secure remote access to the corporate´s applications for users over any device or network resources [14]. FirePass provides endpoint security, good performance, policy management, availability, and scalability. FirePass supports client-server paradigms, and enables a client´s application to communicate to enterprise´s application servers through a secure connection between the browser and the FirePass appliance. It supports Linux, Macintosh, and Windows operating systems. FirePass uses HTTPS protocol with SSL as the transport, so the appliance works through all HTTP proxies including public access points, private LANs, and over public networks. Figure 5 shows FirePass SSL VPN deployment.

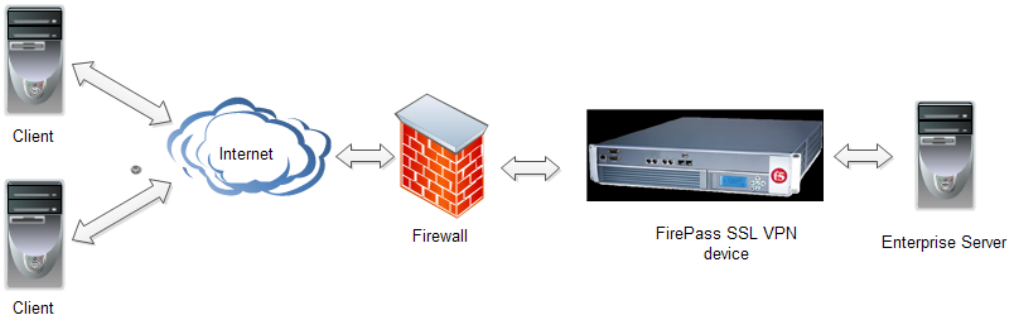

Figure 5: FirePass SSL VPN deployment

University of Tsukuba Virtual Private Network (UT-VPN)

University of Tsukuba Virtual Private Network, UT-VPN is an open source software application that creates secure point-to-point or site-to-site VPN connections. It uses SSL/TLS security for encryption and is capable of traversing network address translators (NATs) and firewalls. It was developed by Daiyuu Nobori, SoftEther Corporation, University of Tsukuba on 2010. UT-VPN uses the OpenSSL library to provide encryption mechanism. It uses username/password for authentication. UT-VPN functions as L2-VPN (over SSL/TLS). It supports UNIX and Windows operating systems.

OpenVPN

OpenVPN is an open source software application that sets up a VPN via an encrypted SSL/TLS connection. OpenVPN allows a secure data connection by using SSL/TLS techniques for authentication and encryption, and does not suffer from the complexity of IPSec VPNs [11]. OpenVPN allows hosts to authenticate with each other using shared private keys, digital certificates or credentials (like username/password). It is installed independently and configured by editing text files manually. The main features of the Open VPN are:

- OpenVPN supports two basic modes, which run at layer 2 and layer 3. It supports non IP packets such as IPX.
- OpenVPN supports multi-client server, where multiple clients can connect to the VPN server through the same port.
- The central firewall can protect the remote device once it has established a VPN tunnel using OpenVPN, even though it is not a local machine.
- OpenVPN has no problems with NAT.
- One port only in the firewall must be opened to permit incoming connections.
- OpenVPN connections can be tunnelled through almost every firewall
- OpenVPN used scripts for several purposes such as authentication, failover and more.
- OpenVPN has proxy support and can be implemented on top of TCP or UDP services.

- Modular design with very simplicity in both networking and security.
- Simple installation and use in any platform.

OpenVPN offers two types of network interfaces using the driver Universal TUN/TAP. It can create either point-to-point layer-3 based IP tunnel (TUN), or a layer-2 based Ethernet TAP that can carry any type of Ethernet traffic. TUN/TAP is a simple structure which increased security compared with other VPN solutions. The Universal TUN/TAP is a virtual network interface that provides support for tunnelling IP traffic. A TUN device is used as a virtual point-to-point interface, and called routed mode. A TAP device is used as a virtual Ethernet adapter, and called bridging mode.

Applications can read and write to this virtual interface, the tunnel driver will use SSL/TLS cryptography to encrypt all data. Then the data will be sent to the other end of the tunnel using either TCP or UDP protocols. Open VPN listens on TUN/TAP devices, gets the data, encrypts it, and sends it to the other VPN end, where another Open VPN process receives the data, decrypts, and delivers it to the virtual network device, which then hands it to the application. Figure 5 shows OpenVPN tunnel connections between two sites.

3.3.3 Including a VPN-based solution into ENCOURAGE

OpenVPN is one of most widely used software among all SSL/TLS VPN solutions, and this has provided OpenVPN with much testing, thus validating it as a mature product. OpenVPN is a good choice for securing ENCOURAGE communications because: i) it is fast, secure and reliable; ii) uses virtual interfaces (TUN/TAP) which they can be accessed without kernel intervention and thus it is more secure to vulnerabilities by design than other solutions; iii) provides portability, and compatibility with NAT and dynamic addresses; iv) it can be used in any platform and is easy to install and configure; v) it is well documented. Figure 6 depicts the main architectural components of the solution.

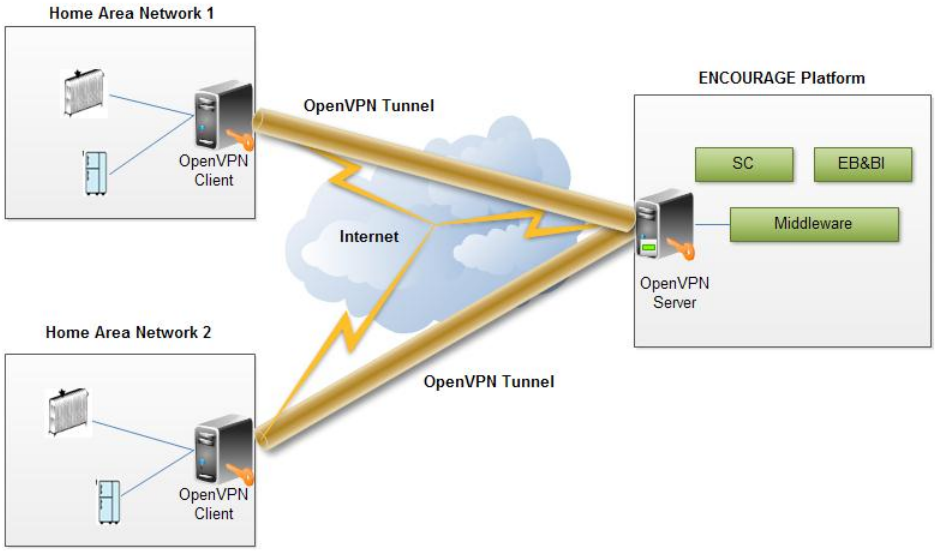

Figure 6: Open VPN architectural components

In this solution, OpenVPN client's configurations in the HAN Gateway are available through a configuration file or a command line interface.

At authentication time, the server can push several configuration data towards the client's sides through the tunnel. The server can tell the clients to use the tunnel as default route to the Internet. Afterwards, all the messages in the client side are routed to the OpenVPN server through the OpenVPN tunnel.

The VPN Server in the ENCOURAGE platform is the component of the OpenVPN architecture that is responsible for authentication, tunnelling, routing, encryption, user management, etc. The server is part of the ENCOURAGE Platform which communicates with the Middleware Plugins.It can also be configured and monitored using a Web GUI. This interface configures most of the functionalities offered by OpenVPN, such as layer 2 or layer 3 routing, server network settings, user permissions authentication and web server certificates. OpenVPN server has also the Connect Client Interface that allows users to connect to the VPN directly through their web browser.

OpenVPN has two authentication modes: A pre-shared static key and SSL/TLS with certificates for authentication and key exchange. In static mode, a pre-shared key is generated and shared between both OpenVPN endpoints before the tunnel is started. Both endpoints will use the same key. In SSL/TLS mode, an SSL session is established with bidirectional authentication. Each endpoint should present its own certificate.

3.4 Security for Smart Energy Profile 2.0 connections

The security solution adopted for the Smart Energy Profile 2.0 implementation in the ENCOURAGE platform will follow the specifications of SEP 2.0 as closely as possible.

As specified by SEP 2.0, securing transactions between the HAN gateways and the ENCOURAGE platform servers is based on using HTTP over TLS [RFC 2818], also known as HTTPS. The TLS version 1.2 [RFC 5246] is to be used and the TLS records are transported using TCP. The TLS handshake mechanism provides mutual authentication based on device certificates or self-signed certificates and TLS records provide encryption and message authentication using the AES-CCM mode of operation. Access control lists (ACLs) control access to resources based on authentication level and client identifiers and address information and a registration list is used for authorizing clients.

3.4.1 Network Access

Each HAN gateway is provided with a Device Certificate that will serve the purpose of authenticating that specific device. All SEP 2.0 certificates are X.509 v3 certificates as defined in RFC 5280. From this Device Certificate, a Device Identifier is generated by applying the SHA256 hash function to the certificate and it is used whenever a client device needs to register in the SEP 2.0 network. Since the Device Identifier is derived from public information (i.e., a Certificate), it can potentially be recreated by an eavesdropper and therefore a 6-digit PIN code shall also be shared out-of-band between the HAN gateway and the ENCOURAGE platform (e.g. at the commissioning phase, in conjunction with the Device Identifier) and the server shall validate that the client has the correct PIN code at registration phase.

Upon a successful authentication, the Device Identifier (in its Short and Long forms) is used to populate the ACL (Access Control List) of one or more resources with a device entry (i.e., whitelist) which ensures only registered devices can access those resources. The default ACL shall be configured in the ENCOURAGE platform for every deployed HAN Gateway.

3.4.2 Resource Access

Once authenticated and authorized through network access, a HAN Gateway can freely communicate in the network at the network layer. However, authentication and authorization at the application layer for accessing resources normally needs to occur to allow hosts to communicate with each other either serving resources as a server host or accessing resources as a client host.

Some public information resources, such as public pricing or public announcements do not require application level authentication and may be accessed simply by using a regular HTTP access, even though this scenario may not be applicable to the ENCOURAGE system.

For resources requiring authentication, the authentication of both hosts, server and client, shall be done as part of the TLS handshake by validating the Device Certificates as described in RFC 5246.

When authenticated, the request is then passed to the ACL (access control list) associated with the resource. The ACL validation allows a granularity of authorization per resource and access method (operation on the REST interface), where each resource maps to a specific SEP 2.0 Function Set.

Additionally, the ACL allows the imposition of restrictions on TCP/IP parameters of the client host (i.e. the IP address and TCP port).

4. Cloud Security

In order to achieve the goal of each use case different cloud services will be deployed over the cloud infrastructure. The cloud infrastructure must grant that those services are used by the authorized stakeholders within each use case, with the required quality of service level. For all the use cases, during the time each use case will take place, there are services that must be working in every moment. Those two requirements, security and availability will determine the way the cloud services must be offered and consumed.

The security required will be enforced by the definition of access policies, secure protocols over an insecure network, and the required specific measures for each use case.

The availability requirements will be achieved first, introducing the required techniques for reduce the non-planned downtime of each service and elaborating a contingency plan in which a risk assessment will analyse the potential menaces and the action related in order to mitigate the effect in case that the menace became real.

4.1 Contingency Plan

For each use case of ENCOURAGE, during the delivery of the cloud services some unexpected issue could occur that interrupt the properly deliver of the service. In this point an analysis of the risks that could affect the use cases, and how to manage it to avoid non planed downtime in the service delivery.

4.2 Risk Assessment

A List of the potential issues that can occur regarding the type of cloud deployment, technological stack, hardware employed, load of the system, network traffic, etc. for each use case, actions to minimize the risk and related action to be performed in case the issue occurs. Impact of the risk assessment in the overall architecture, for example fail over requirements.

RISK	EFFECT	CONTINGENCY PLAN	CLOUD SERVICE
Unauthorized access to services	Random behaviour Data modification Confidentiality of data	Enforce access polices and secure protocols, physical security.	IaaS/SaaS/PaaS
Software failure	Overall system malfunctioning, service downtime	Keep update the software from provider specification	IaaS/SaaS/PaaS
Blackout	Non planned	Distribute the	IaaS

Natural disaster(earthquake, floods, fire,…)	service downtime Data corruption	infrastructure over different physical centres	
Hardware failure	Service downtime. Data corruption	Eliminate single point of failure within each deployment	IaaS/SaaS/PaaS
Network failure	Service downtime. Data corruption	Search for alternative network provider	
Data corruption	Wrong reports and results. Overall error conclusions.	Backup and recovery policy.	

Table 2: Risk Assessment

4.3 Availability

The main quality required for the cloud services to be deployed for the ENCOURAGE project is the availability of the services, most of them will demand that the required cloud services will work properly regarding each Service Level Agreement (SLA) related with each cloud service deployed. Once the risk assessment has been done and regarding the contingency plan herein is a depiction of the different actions to be addressed from the security point of view, in order to grant the required availability for each deployed service within each use case.

4.3.1 Distributed Data Centre

The data centres are the physical facilities where the infrastructure is allocated. If the cloud services are deployed over a single data centre, in case of failure of the data centre the whole cloud services will fail. To avoid this situation, more than one data centre will be in charge of providing physical infrastructure where the cloud services can be deployed. A duplication of the physical infrastructure will reduce the risk of overall downtime.

The solution proposed is to devote at least two physical servers with the required capacity regarding the capacity plan. This servers will be distributed physically either within the same premises either separated premises in order to reduce the risk of failure.

The centralized monitoring of the health of the physical infrastructure, based in Zabbix, will provide a proactive management of the infrastructure, reducing the possibilities of non-planned downtime related with the data centre.

4.3.2 Elimination of single point of failure

A single point of failure is a component of a cloud service that in case of failure will produce the cloud service to fail. Therefore to avoid this scenario an analysis of the single points of failure ought

to be done, and once identified eliminated. The next techniques are aimed to avoid single points of failure within the cloud services deployed.

4.3.2.1 Load Balancer

Different techniques to distribute the incoming load can be applied regarding the platform required to deploy a cloud service. A fail over strategy of the load balancer related to each service must be addressed. This can be done providing additional load balancer for redundancy purposes.

4.3.2.2 Storage Services

In order to avoid single points of failure for the storage services, a dedicate network that provides storage services must be implemented. The storage services will provide redundancy and an effective disaster recovery process.

4.3.2.3 Database Services

The database services provide logical access to related data. Failover procedures will be enforced regarding the database engine chosen for each cloud service. An active/passive or active/active database deployment will eliminate any SPoF related with data base services.

4.3.2.4 Redundancy of Components

Independently of the platform or programming model chosen for the cloud services deployment to avoid SPoF a redundancy of components will reduce the risk of a total blackout due to a single failure. In order to take advantage of this strategy a configuration management process is to be done, this will assure that all the components will work and produce the same expected results.

4.3.2.5 Monitoring Services

The monitoring of the different components will provide the required information in order to take the appropriate decisions regarding the deployment. Real Time information of the status of each component within a cloud service will be provided by the monitoring tool Zabbix.

4.3.3 Assets Management

The appropriate management of the assets that comprises a cloud service would allow to control effectively the health of the service, and to grant the proper functioning of cloud service deployed.

4.3.3.1 Assets Inventory

Comprehensive information of all the assets involved in the deployment of a cloud service will be available for the cloud service owner or manager.

4.3.3.2 Assets Updates

To avoid problems related with bugs and in order to provide the latest functionalities of the different assets of a cloud service, an appropriate update policy has to be enforced. With the information provided from the assets inventory.

4.3.4 Back-Up Recovery Policy

Depiction of the backup and recovery procedures and protocols when a major issue occurs and the service has been interrupted and is required to be re-established. It ought to involve data storage, Virtual hardware failure, Hardware failure, among others.

4.4 Access Control Policies

The model of control model specifies who is allowed to perform what kind of operations on content under certain conditions. The model defined have to assure the integrity and the confidentially of the data.

4.4.1 Identity Management

In order to avoid unauthorized access to any cloud service or data within the ENCOURAGE framework a reliable identity management policy must be enforced.

Each use case is defined by some scenario and some actors and stakeholders. These actors will consume the deployed cloud services and will produce the data employed for the expected conclusions for each use case. The identity management of each actor has a life cycle which includes:

- Creation of actor`s identity within the ENCOURAGE`s cloud services
- Role management for enforcing effectively security policies
- Creation and modification of credentials
- Permission´s management
- Credential´s verification
- Grant or refuse access to a cloud service

In order to address this requirements to tackle the risk of unauthorized access to the cloud services within the ENCOURAGE framework two services will be offered, OpenLDAP and JOSSO. This would cover the identity management for the required cloud services.

4.4.1.1 OpenLDAP

OpenLDAP is an implementation of the Lightweight Directory Access Protocol. It offers a directory service, where namespaces are stored. The namespace is composed by all the information required to identify any object within the namespace.

LDAP is a protocol for accessing directory services, specifically X.500-based directory services.

With the deployment of this service within the ENCOURAGE´s cloud service the following features could be addressed:

- Machine Authentication
- User authentication
- Users/systems groups and roles
- Application Configuration Store
- Use case organization representation.

4.4.1.2 JOSSO

JOSSO stands for Java Open Single Sign On, and is SSO solution for web applications. It is an open Source Java EE based software for user authentication and authorization. The framework allows multiple web servers/applications to authenticate users with credential store.

The main features that JOSSO will provide to the ENCOURAGE cloud services are:

- J2EE, SPRING AND WINDOWS transparent cross-domain/cross-organization SSO.
- SAML support for seamless Internet/Federated SSO
- Supports Strong authentication using X.509 client certificates
- Runs in:
  - Apache Tomcat
  - JBoss
  - Bea Weblogic
  - Websphere
  - Apache Geronimo
  - Windows IIS
- Out-of-the box compatibility with Alfresco CMS, OpenCMS
- Native Apache Https 2.x support thus enabling transparent SSO with Ruby, PHP, Perl applications.

- Includes virtual directory functionality for allowing authentication against multiple disparate identity silos.
- Communicates with credential stores over the LDAP or JDBC connection.
- SSO services are exposed using SOAP over HTTP protocol allowing it to integrate with non-Java applications.
- Standard Based: JAAS, Web Services/SOAP, EJB, Struts, Servlet/JSP, J2EE.

4.5 Communications

The requirements for all the types of communications are addressed here, with a special focus on the publish/subscribe model used by the message broker. Probably additional topics must be included regarding some specific issues for de probes of each use case

4.5.1 Network security configuration

Each different business case of ENCOURAGE will take place over a different physical scenario, but will access to the same type of cloud services, identity management, load balancer services, front end services, and high availability services, etc. That provides a division within which is public and which is not. In order to separate the private area from the public a set of firewall will create a DMZ.

Within that DMZ two additional areas will be created, the first area will offer the identity management services and the front end services to authorized users from the public network. The second area will be devoted to deploy the cloud services in the different cloud level (IaaS, SaaS, PaaS).

The access to each area is controlled at two levels, physical and logical. The physical level will be provided by firewalls and the logical level will be provided by the identity management that includes directory services and Single Sign On services.

Regarding the requirements of each particular business case, additional VPN would be created as well as additional firewall configurations.

4.5.2 Protocols for communication over an insecure network

As a starting point it will be assumed that the actors involved within a business case will access to the ENCOURAGE cloud services through an insecure public network, therefore it is required to enforce the use of secure communication protocols in order to assure the confidentiality of the information transmitted. TCP/IP will be the protocol employed for communications over a public network with SSL v.3.0 as secure layer to assure safety communicating over public network

4.5.3 Publish/Subscribe model for message broker

Depiction of the model for messaging employing the publish/subscribe paradigm. Application of this model to the ENCOURAGE project regarding each use case.

4.5.3.1 Availability

Cloud services provide availability of data. It is possible that a subscriber registers to a broker by specifying subscription predicates. The broker is responsible for delivering any event that satisfies such predicates to the subscriber within a pre-defined period of time. If any component of the broker fails along the event delivery path should not prevent the subscriber from deliver this event. So either the event delivery route be reconfigured dynamically or multiple event delivery routes should be established. Regarding the size of the pub/sys, it will take more time to propagate subscription information into the whole network. Some KPI ought to be defined in order to create a reliable system, number of single points of failure, registration information propagating time, and others.

Solutions to provide the availability regarding each use case of encourage ought to be depicted here, as well as an analysis of the main KPIs defined previously of each solution.

4.5.3.2 Confidentiality

Only authorized subscribers could access to the events. The policies for access policies should also be applied in this point with special focus on the key management and an encryption/decryption scheme. The event should be encrypted when it is delivered to subscribers, so that only authorized subscribers are able to decrypt it. A study of the number of keys shared by the universe of subscribers ought to be done in order to choose the key type as well as the adequate compromise between the number of subscribers, the number of group members and volume of published events.

4.6 Encourage Secure Architecture Proposal

Within the ENCOURAGE project, four business cases will be developed. Each of those BC will implement many different use cases aimed to achieve the specific energetic goals regarding each scenario. Those use case will employ the cloud services offered regarding the technical possibilities of the actors involved, probes, gateway, protocols available etc.

The following diagram offers a complete schema of the security offer for any of the ENCOURAGE business case that will be deployed. In the schema is shown the network configuration.

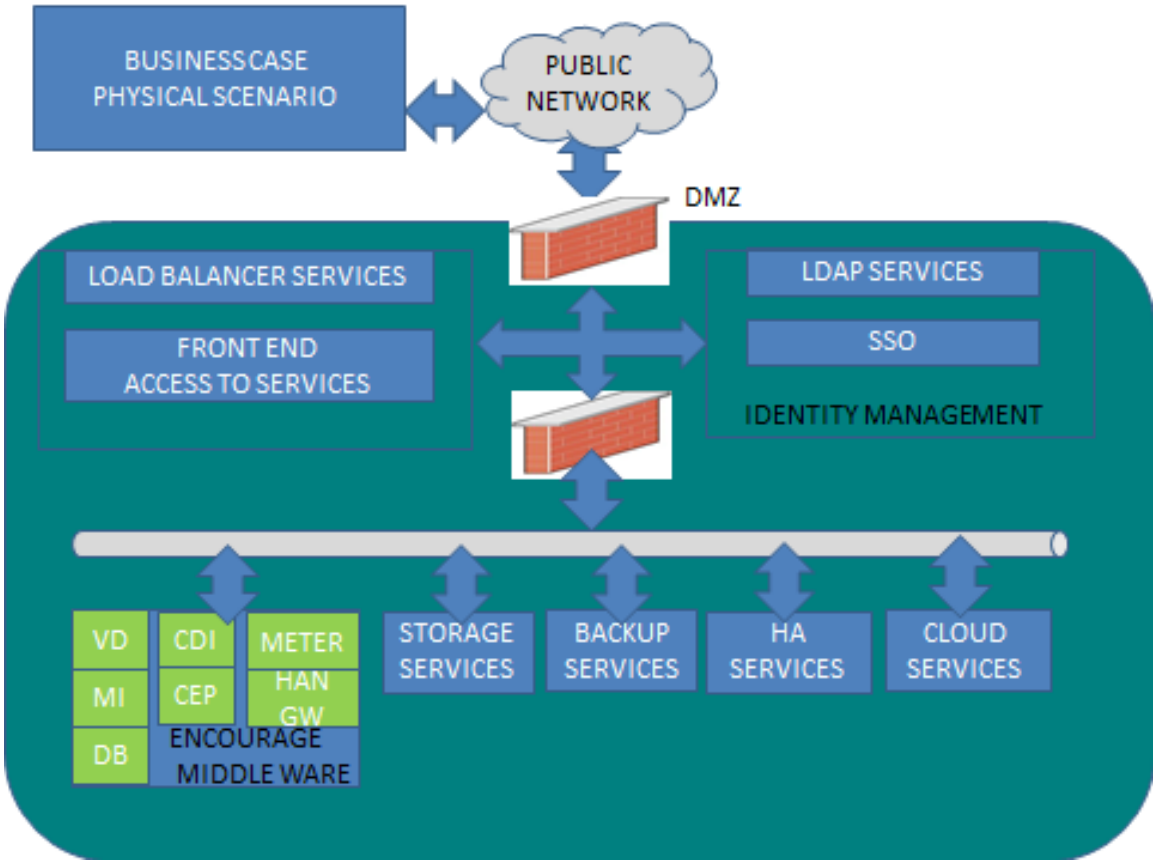

Figure 7: Encourage Security Architecture

5. Middleware Messaging Security

The ENCOURAGE Middleware constitutes the infrastructure which provides the core of the platform.

The Messaging Infrastructure links all higher level ENCOURAGE modules. This solution allows reducing the complexity of inter-application communications since it works as a message bus, thus applications use only one communication protocol and are only required to maintain one connection with the message bus. This Infrastructure should be capable of supporting different types of communication paradigms: publish/subscribe and also request/response.

This requirements are addressed using RabbitMQ, an open source message broker software (i.e., message-oriented middleware) that implements the Advanced Message Queuing Protocol (AMQP) standard.[1]

5.1 RabbitMQ security

5.1.1 Access Control

When an AMQP client establishes a connection to an AMQP (Advanced Message Queuing Protocol) server, it specifies a virtual host within which it intends to operate. A first level of access control is enforced at this point, with the server checking whether the user has any permission to access the virtual hosts, and rejecting the connection attempt otherwise.

Resources, i.e. exchanges and queues, are named entities inside a particular virtual host; the same name denotes a different resource in each virtual host. A second level of access control is enforced when certain operations are performed on resources.

RabbitMQ [15] distinguishes between *configure*, *write* and *read* operations on a resource. The *configure* operations create or destroy resources, or alter their behaviour. The *write* operations inject messages into a resource. And the *read* operations retrieve messages from a resource.

5.1.2 Default database access

When the server first starts running, and detects that its database is uninitialized or has been deleted; it initializes a fresh database with the following resources:

- a virtual host named /
- a user named guest with a default password of guest, granted full access to the / virtual host.

5.1.3 SASL Authentication

RabbitMQ has pluggable support for various SASL authentication mechanisms. There are three such mechanisms built into the server: PLAIN, AMQPLAIN, and RABBIT-CR-DEMO, and one - EXTERNAL - available as a plugin. You can also implement your own authentication mechanism by implementing the rabbit_auth_mechanism behaviour in a plugin.

5.1.4 Keys, Certificates and CA Certificates

OpenSSL is a large and complex topic. For a thorough understanding of OpenSSL and how to get the most out of it, we would recommend the use of other resources, for example Network Security with OpenSSL.

OpenSSL can be used simply to establish an encrypted communication channel, but can additionally exchange signed certificates between the end points of the channel, and those certificates can optionally be verified. The verification of a certificate requires establishing a chain of trust from a known, trusted *root* certificate, and the certificate presented. The *root* certificate is a self-signed certificate, made available by a *Certificate Authority*. These exist as commercial companies, and will, for a fee, sign SSL Certificates that you have generated.

5.1.5 Levels of Trust

When setting up an SSL connection there are two important stages in the protocol.

The first stage is when the peers *optionally* exchange certificates. Having exchanged certificates, the peers can *optionally* attempt to establish a chain of trust between their root certificates, and the certificates presented. This acts to verify that the peer is who it claims to be.

The second stage is where the peers negotiate a symmetric encryption key that will be used for the rest of the communication. If certificates were exchanged, the public/private keys will be used in the key negotiation.

Thus you can create an encrypted SSL connection *without* having to verify certificates. The Java client supports both modes of operation.

6. Conclusions

The Communication of the ENCOURAGE reference platform involved many of the modules and plugins defined in the Architecture. The Communication is a fundamental task in Encourage, since there is an important quantity of data monitored and evaluated. The management of information must carefully accomplish security restrictions. Some of the procedures described in this document might be updated according to the necessities that will come up along the implementation of the Encourage Communication's Architecture.

That work done will ensure the project requirements defined in wp2, the Architecture, and also to fit with project constraints.

References

[1] Potlapally, Nachiketh R., et al. "A study of the energy consumption characteristics of cryptographic algorithms and security protocols." *Mobile Computing, IEEE Transactions on* 5.2 (2006): 128-143.

[2] Fujisaki, Eiichiro, and Tatsuaki Okamoto. "Secure integration of asymmetric and symmetric encryption schemes." *Advances in Cryptology—CRYPTO'99*. Springer Berlin/Heidelberg, 1999.

[3] Garfinkel, Simson. *PGP: pretty good privacy*. O'Reilly Media, Incorporated, 1994.

[4] Rescorla, Eric. *SSL and TLS: designing and building secure systems*. Vol. 1. Reading, Massachusetts: Addison-Wesley, 2001.

[5] Kent, Stephen. "IP encapsulating security payload (ESP)." (2005).

[6] Kent, Stephen. "IP authentication header." (2005).

[7] Harkins, Dan, and Dave Carrel. *The internet key exchange (IKE)*. RFC 2409, November, 1998.

[8] Yuan, Ruixi, W. Timothy Strayer, and Tim Strayer. *Virtual private networks: technologies and solutions*. Addison-Wesley, 2001.

[9] RabbitMQ. http://www.rabbitmq.com/, last accessed on 2012/11.

[10] ZigBee Smart Energy Profile Specification SEP 2, Revision 16, version 1.1, ZigBee Alliance, March, 2011.

[11] M. Feilner, OpenVPN: Building and Integrating Virtual Private Networks, Packt Publishing, 2006.

[12] XU, Jia-zhen, and Shen-meng CHEN. "Comparison and analysis of IPSec-based and SSL-based VPN [J]." Computer Engineering and Design 4 (2004): 031.

[13] Barrcuda SSL VPN, https://www.barracudanetworks.com/ns/products/sslvpn_overview.php, last accessed on 2012/11.

[14] Firepass SSL VPN, http://www.f5.com/products/firepass/, last accessed on 2012/11.

[15] http://www.rabbitmq.com.