



The E-Safety Network & the Emergency Provider Access Directory



888 17th St, N.W., Washington, DC 20006
www.comcare.org

Copyright, 2004. The ComCARE Alliance

TABLE OF CONTENTS

1 INTRODUCTION.....	1
1.1 OVERVIEW OF THE E-SAFETY NETWORK	1.1
1.2 OVERVIEW OF EPAD	1.2
1.3 USES OF THE EPAD	1.3
1.4 GETTING EPAD DONE.....	1.4
2 E-SAFETY NETWORK COMPONENTS.....	2
2.1 EPAD	2.1
2.2 AGENCY ALERTS	2.2
2.3 SIGNAL CLEARINGHOUSE.....	2.3
2.4 INCIDENT WEBSITES	2.4
2.5 INTELLIGENT MESSAGE BROKER	2.5
2.6 SIGNAL SOURCES	2.6
2.7 ADMINISTRATION	2.7
3 INTELLIGENT MESSAGE BROKER.....	16
3.1 INTERFACES	3.1
3.2 NOTIFICATION SERVICE	3.2
3.2.1 <i>Alert Push Services</i>	3.2.1
3.2.2 <i>Alert Query</i>	3.2.2
3.2.3 <i>Signal Notification Handoff</i>	3.2.3
3.2.4 <i>Signal Source</i>	3.2.4
3.2.5 <i>Signal Query</i>	3.2.5
3.3 SENDER CONTROL	3.3
3.4 RECEIVER CONTROL	3.4
4 NETWORK DISTRIBUTION.....	4
4.1 INDEPENDENT, PARALLEL EPADS.....	4.1
4.2 CENTRAL EPAD WITH LINKED CHILDREN	4.2
4.3 INDEPENDENT, INTERLINKED EPADS	4.3
4.4 INTERLINKED EPAD HIERARCHY	4.4
4.5 HYBRID APPROACHES	4.5
5 SECURITY CONSIDERATIONS	5
5.1 AUTHENTICATION AND NETWORK SECURITY	5.1
5.1.1 <i>SOAP/HTTP Authentication</i>	5.1.1
5.1.2 <i>Validating Identities</i>	5.1.2
5.1.3 <i>Protecting Data in Transit</i>	5.1.3
5.2 ROLES AND PRIVILEGES	5.2
6 DATA STORAGE.....	6
6.1 EPAD ENTRY CHARACTERISTICS	6.1
6.2 PERFORMANCE ISSUES	6.2
6.3 IMPORT FROM EXISTING SOURCES	6.3
7 SCALABILITY STRATEGIES	7

7.1	ASSESSMENT OF SCALABILITY CHALLENGES	7.1
7.2	HIGH LEVEL TECHNICAL APPROACH	7.2
8	DATA INTEGRITY	8
8.1	DATA REVALIDATION AND HOUSEKEEPING	8.1
8.2	ADMINISTRATIVE MESSAGES	8.2
8.3	DATABASE DESIGN	8.3
8.4	MESSAGE QUEUES	8.4
8.5	BACKUP AND RECOVERY	8.5
8.6	AUDITING	8.6
9	REPORTING	9
10WEB SERVICES OVERVIEW.....	10
10.1	SERVICE-ORIENTED ARCHITECTURE.....	10.1
10.2	SOAP	10.2
	<i>10.2.1 Overview</i>	<i>10.2.1</i>
11	SAMPLE EPAD IMPLEMENTATION.....	11
11.1	EPAD ARCHITECTURE	11.1
11.2	API DESIGN	11.2
	<i>11.2.1 Operations</i>	<i>11.2.1</i>
11.3	DATA STRUCTURES.....	11.3
	<i>11.3.1 Documentation of Data Structures</i>	<i>11.3.1</i>
	<i>11.3.2 Sample EPAD Data Structures</i>	<i>11.3.2</i>
11.4	STUB GENERATION	11.4
11.5	AUTHENTICATION	11.5

1 Introduction¹

1.1 Overview of the E-Safety Network²

Best practices in network technology enable us to integrate and accelerate our emergency communications in ways never possible before. The E-Safety Network initiative, spearheaded by the national non-profit ComCARE Alliance, is demonstrating and deploying “interoperability now” for computerized information systems in emergency response, emergency management and homeland security.

The E-Safety Network is not a product. It is an architecture, a web services system based on open, non-proprietary technical standards that ensure technical interoperability between systems and agencies, and encourage the development of new applications, competing for a national market. It uses a services oriented architecture. In this way it secures for emergency response agencies the benefits of reliability, efficiency and continuing innovation through open market competition among solution providers that our commercial IT markets have seen in the last two decades.

1.1.1 Background

Private industry has developed and proven standard methods for linking diverse data systems into a coherent national and global architecture. One of the most significant tools embraced by highly networked companies over the past decade is a data format called XML, which has emerged as the information industry’s preferred method for exchanging data between computer systems. But the XML technology, familiar to private-sector information technology providers, has yet to be systematically applied in public-safety data communications. Similarly, packet switching and Voice over Internet Protocol (VoIP) are become rapidly diffused in commercial markets, but not in emergency response.

Although Geographic Information System (GIS) technology has been adopted widely for data display and analysis in emergency response and emergency management applications, it has yet to be broadly applied in the emergency communications infrastructure in an integrated way. Remarkably, there is at present no shared national online directory for instantly identifying which first-response and emergency support agencies have responsibility at any particular geographic location and routing time-critical information to them.

¹ Copyright, 2004. The ComCARE Alliance. This paper is an amalgam of prior conceptual thinking and writing on both the E-Safety Network and EPAD by ComCARE staff, with new analysis and contributions by the staff of Battelle Memorial Institute. Those individuals who contributed the most to it include: current and former employees of ComCARE: Patrick Halley, Stephen Seitz, Sukumar Dwarkanath, Art Botterell, and David Aylward; and Battelle staff, Joseph Kessler and Gary Ham.

² To provide open and participative guidance of the E-Safety Network initiative and to identify and pursue other activities to improve emergency response and homeland security, including commercial deployment opportunities, the ComCARE Alliance has established a national “Safety Technology Initiatives Committee” (STIC) with representatives from industry and public-safety agencies and organizations. Members of STIC have committed themselves to open, non-proprietary networks for emergency response, and agreed to a code of professional ethics. For more information on the E-Safety Network Initiative, contact: The ComCARE Alliance, Safety Technology Initiatives Committee, 888 17th Street NW, 12th Floor, Washington, D.C., 20006, (202) 429-0574, <http://www.comcare.org>, sdwarkanath@comcare.org.

The ComCARE Alliance, a six year old not-for-profit coalition of over 90 organizations in the medical, emergency response and management, 9-1-1, telecommunications, transportation and technology sectors, began to engage these problems of the emergency communications and IT infrastructure as it sought to encourage the deployment of consumer safety technologies, particularly Enhanced 9-1-1 for wireless telephones and automobile “telematics”. In 2000, for example, the Final Recommendations of the National Mayday Readiness Initiative (NMRI) sponsored by ComCARE and the US Department of Transportation called for a wide variety of steps to upgrade the nation’s emergency response communications and information technology infrastructure so it could handle critical data from the next generation of in-vehicle telematics systems offered by firms like OnStar and ATX.

From this and other intensive exploratory initiatives around the country with a wide diversity of emergency leaders, ComCARE members identified a larger need for a national data communications and information technology architecture based on open, non-proprietary standards and a shared, “spatially aware” directory of network users. At the request of its members ComCARE facilitated the development of a Vehicular Emergency Data Set (in XML) by a team of emergency medical, 9-1-1, and technology organizations. OnStar and ATX plan to use this data format. Similarly, ComCARE financially supported the development of the Common Alerting Protocol (CAP), and was the first organization to use it in field trials. CAP is the first nationally approved XML emergency communications standard.

The terrorist attacks of September 11, 2001, highlighted the critical need for “interoperability” in emergency communications. Until now that term’s use has been confined almost entirely to the realm of voice two-way radio systems used by field first-response agencies for at-scene communications. But the interoperability challenge extends far beyond two-way radios and the scenes of incidents.

The same problem arises when different agencies’ information systems – computer-aided dispatch systems, emergency-management information systems, public health systems, wireless data systems in the field, and many others – need to exchange up-to-the-minute information. The problem of regional, state or national communications is particularly acute because, remarkably, many of the nation’s 80,000 emergency agencies still have little to no data communications capability at all.

1.1.2 The E-Safety Network In Action

In 2000 and 2001 ComCARE assembled a team of emergency response leaders and innovative companies to detail a new kind of information architecture, and to demonstrate how diverse commercial technologies and products could be integrated into an E-Safety Network. Supported in part by a grant from the Virginia Department of Transportation, these ideas and then products were tested with emergency experts in the Northern Shenandoah Valley of Virginia. The consortium developed, and in 2002 and 2003 deployed and demonstrated, an open, standards-based network that provided responders with immediate life-saving information, on hazardous materials incidents, medical and mass-casualty events, and other situations involving emergency data-sharing, alerting and notification, not just vehicle crashes.

In 2003 this system was tested in several multi-agency emergency exercises including a full-field terrorism and hazardous materials exercise; a regional, federally-funded smallpox bio-

terrorism exercise involve hospitals and public health districts; a state pneumonic plague exercise, and a five county exercise involving multiple disasters and “garden variety” emergencies.

In all these exercises, simulated and actual data from a number of sources were relayed instantly to computers at emergency response dispatch and emergency medical facilities, and via various wireless devices to first responders and leadership staff in the field. Geographic locations of incidents were automatically plotted on a shared online map, providing a “common operational picture” for all players and providing instant “single-click” access to detailed incident information and environmental and historic data.

“The key is keeping people connected both ways, which is a real networking challenge,” said Jack Potter, MD, Director of Emergency Medicine for Valley Health System (which includes Winchester Medical Center and three other hospitals in the Shenandoah Valley) and a key leader and proponent of the system. “Getting the data we want into the network and available for decision-making early on is a big part of the project.”

In some of the exercises the E-Safety Network was expanded to deliver “streaming” video. Short, pre-recorded training modules developed by state health officials provided first responders and health professionals with just-in-time and short refresher training on the simulated smallpox threat. A live, secure video briefing streamed to participants from a federal studio in Washington, D.C., added to the scope and realism of the exercise.

“We have the core infrastructure deployed to create a test bed for new applications,” Dr. Potter says. “We are actively seeking more corporate partners which want to refine their emergency applications, using our willing, educated and involved safety leaders. The initiative will expand and grow with different applications being added. It's just like the Internet to start with. If it's an open environment that everybody is allowed to plug into, it works best for everybody. That's what our country did with open standards computing, including the Internet. We want to do the same thing here for emergency communications.”

1.1.3 Elements of the E-Safety Network

The E-Safety Network is a unified emergency web services information architecture which ties together the various law, fire, emergency medical, public health, transportation and emergency-management/homeland security agencies. It consists of:

- Reliable and secure broadband data connections using Internet protocols;
- Secure web services message-routing systems based on a standard suite of non-proprietary XML message formats using SOAP “packaging” for both specialized and general emergency data exchange;
- A shared national and non-profit Emergency Provider Access Directory (EPAD). This is a comprehensive real-time database of emergency and allied agencies that enables automatic routing of messages to responsible agencies based on their geographic areas of responsibility and their incident information requests (established and maintained by the agencies themselves through an on-line process);

- Published interfaces to a wide variety of existing data sources (sensors, vehicle tracking systems, dispatch systems, wireless E9-1-1, etc.) and users (dispatch centers, treatment and response agencies, incident-management and GIS systems, “situational awareness” displays, emergency alerting systems, etc.);
- Introduction of innovative technologies for data collection, analysis and presentation, both in command centers and in the field;
- A body of proven “best practices” for building the national E-Safety network, including access and security policies, interface validation and certification processes, and data quality and format standards, and;
- A cooperative, multi-vendor effort to demonstrate, evaluate and improve the E-Safety Network through an ever-expanding series of real-world implementations.

Our commitment to national data and interface standards is the critical corollary to our strong belief that individual agencies and localities should be empowered to determine how best to use these new information sources and tools in their communities. We can no longer afford information “silos” -- but local and state innovation and flexibility in how information is received, processed, and acted on must be enhanced.

1.1.4 Goals for the E-Safety Network

The private and public sector participants in the E-Safety Network have committed themselves to:

- Develop a new kind of emergency information architecture, based on best practices developed and proven in the private sector;
- Develop and demonstrate a scalable, national emergency communications capability based on open standards that can be implemented at reasonable costs;
- Develop and demonstrate recommended non-proprietary standards for specialized functions (automatic vehicle crash notification, hazardous materials transportation accident alerts) and general emergency alerts, notifications and information exchange;
- Provide education and outreach to aid government agencies and private-sector partners in leveraging these new technical opportunities, and;
- Create a framework for free-market competition in the application of improved standards-based products for emergency and public-safety communication.

1.2 Overview of EPAD

The Emergency Provider Access Directory (EPAD) initiative is an effort to organize contact information for key emergency providers and provide a measure of integration not currently available. There is currently no accurate, comprehensive printed or electronic directory of telephone numbers, Internet addresses, street addresses and other relevant contact information for all federal, state and local emergency response agencies in the U.S. This is a critical missing link in the infrastructure of the public safety community in America that

impairs the ability of these agencies to respond to individual or mass emergencies. A fundamental infrastructure capable of receiving and disseminating voice and data communications to and among multiple emergency responders is badly needed.

The intent of the EPAD initiative is to fill this surprising gap in emergency preparedness. The purpose of this document is to evaluate, review, and make specific technical and organizational recommendations concerning the technical architecture of EPAD and its connection and use in the associated E-Safety Network.

1.3 Uses of the EPAD

The EPAD can help to address an obvious gap in emergency response. When a tornado, hurricane, terrorist threat, or event involving a weapon of mass destruction emerges, there is no way for the President, the Department of Defense, Department of Homeland Security, state Governor, or other public safety official to quickly alert all appropriate emergency responders and agencies to the emergency. Similarly, a 9-1-1 center in a particular state may not have a simple way to contact another 9-1-1 center in a neighboring state. This is true for both voice and data communications. In addition, it may also serve to improve the capabilities of select private companies. National private call centers serving the exploding telematics market and wireless telephone operators also need an accurate, comprehensive, and electronic database that contains reliable emergency contact information. When the OnStar call center in Detroit, MI receives information on a crash or stolen vehicle in Billings, MT, they need accurate information on the appropriate emergency responders to contact in Billings. Today this involves a phone call and the relay of available information via telephone conversation. The EPAD will enhance the voice-centric communications system existing today by enabling the call center to electronically send all available data on an incident to emergency responders.

In all such applications, the net effect of the EPAD initiative – along with the E-Safety Network infrastructure – should be to make emergency responses more accurate, responsive, and reliable than is currently possible.

1.4 Getting EPAD Done

Implementing EPAD will require a concerted technical initiative. However perhaps a bigger challenge may be the impact on the organizations that it aims to serve. The creation of a directory will require a coordinated effort in each state. The Governor and other political leaders will need to bring all parties involved in emergency response together in a state to register their information into the directory. In so doing, we will be starting a process of statewide coordinated planning for emergency communications as encouraged in the Wireless Communications and Public Safety Act of 1999. Through several partnerships, the ComCARE Alliance has initiated planning sessions in several states to date and is prepared to continue efforts at the local, state and national level to complete and manage the directory. Once created, the EPAD will be supported by subscriptions from telecommunications carriers, public safety organizations, federal agencies, telematics service providers, home security companies and others.

1.5 Summary of Overall Requirements and Related Considerations

EPAD will provide (or enable) the following functionality for the responder community:

- Ability to specify the type of information that is to be automatically provided to qualified parties on a “need to know” basis as authorized by appropriate state or local officials. The manner in which this will be specified and filtered can be implemented in a variety of ways, and must be decided upon based on best practices.
- List appropriate emergency contact information for private entities which are able to notify the public at large. Currently, most major public warnings are delivered in the form of television or radio broadcasts. However, new communications technologies are developing the capability to deliver effective localized emergency warnings. Wireless carriers and pagers, short message services (SMS), instant message services, automotive telematics services, and Internet services could all be utilized in the future to alert subscribers within a localized area or on a national scale in real time.
- Provide a process to collect, update, and connect by modern telecommunications technology all existing emergency contact data. An efficient means of capturing timely and accurate contact information is crucial to the usefulness of the service. Recognition of EPAD as an authoritative and accurate source of critical information is paramount to its acceptance and a key driver in achieving a “critical mass” that motivates responders to participate actively.
- Maintain all information on secure network servers. The security will be consistent with the best commercial and government practices and provide all necessary steps to erect firewalls and protect the database from hackers.
- Provide a wide variety of modern telecommunications and commuting applications which will be offered on a competitive basis by public and private entities.
- A formal process has begun to involve the various emergency response communities in the next phase of the technical and operational development of EPAD.
- EPAD will remain vendor neutral. This is accomplished by insisting that all interactions with the E-Safety Network occur through an open, standards-based interface. This will produce a more modular and nimble network that can adjust to changing circumstances.
- We are now determining functional requirements for the EPAD API. In addition to supporting operations on agency contacts, these requirements must also account for security and permissions, reporting, auditing, and querying.
- Since alert routing will be very flexible, and EPAD will offer an interface for performing routing queries, the query language (or format) to be used for this is being worked out.
- Representation, storage, and handling of jurisdictional and “interest area” boundaries are being designed.
- Interface protocols must be settled upon for each service exposed by the E-Safety Network. In most cases, the protocol will be SOAP/XML over HTTP. In some cases, however, secondary interfaces may be offered for various reasons. One example is support for HTTP/GET as an addition alert polling protocol, since it is

somewhat easier to leverage in some cases than SOAP/XML. Another example is using the “hybrid pull” polling approach that uses a secondary channel to notify agencies when they need to poll the network for alerts.

- An application server platform must be selected that accommodates modern server computing features such as clustering. These servers are available on multiple platforms such as Java (J2EE) and Microsoft .Net.
- The IMB will likely employ the use of message queues to hold alerts for agencies between the time they are generated and the time they are transmitted to those agencies. There are many existing message queuing solution, though it may also suffice to store the alerts in a transient table. The lifespan of these transient alerts (such as storage duration) must be examined and channeled into the proper technology.
- The mechanism by which alerts are retrieved from the IMB can have a great impact on the cost and effort required by agencies to leverage the E-Safety Network.
- The use of strong encryption, such as SSL, for security during transport is highly recommended. This will provide a degree of privacy similar to that used on financial web sites.
- The list of possible roles by registrants will be defined more fully, as well as the privileges that can be assigned to each role.
- All security rules should be implemented on the server side for greater security and control (as opposed to an all-too-common practice of hard-coding them into client software). In this way, the server becomes a “black box” whose functionality cannot easily be hacked, and also allows the security model to be updated while affecting only server-side code.
- We are designing the service with scalability in mind from the start. We assume that all software must support clustered environments, which allow scalability to be increased by adding more server hardware rather than requiring a reengineering effort.
- We are carefully considering how geographically oriented queries will be handled. Routing queries can be quite data intensive when they refer to jurisdictional boundaries. We are exploring the use of a strategy of filtering on other, lower cost criteria first in order to avoid computations wherever possible.

2 E-Safety Network Components

The following diagram depicts the general layout of the E-Safety Network:

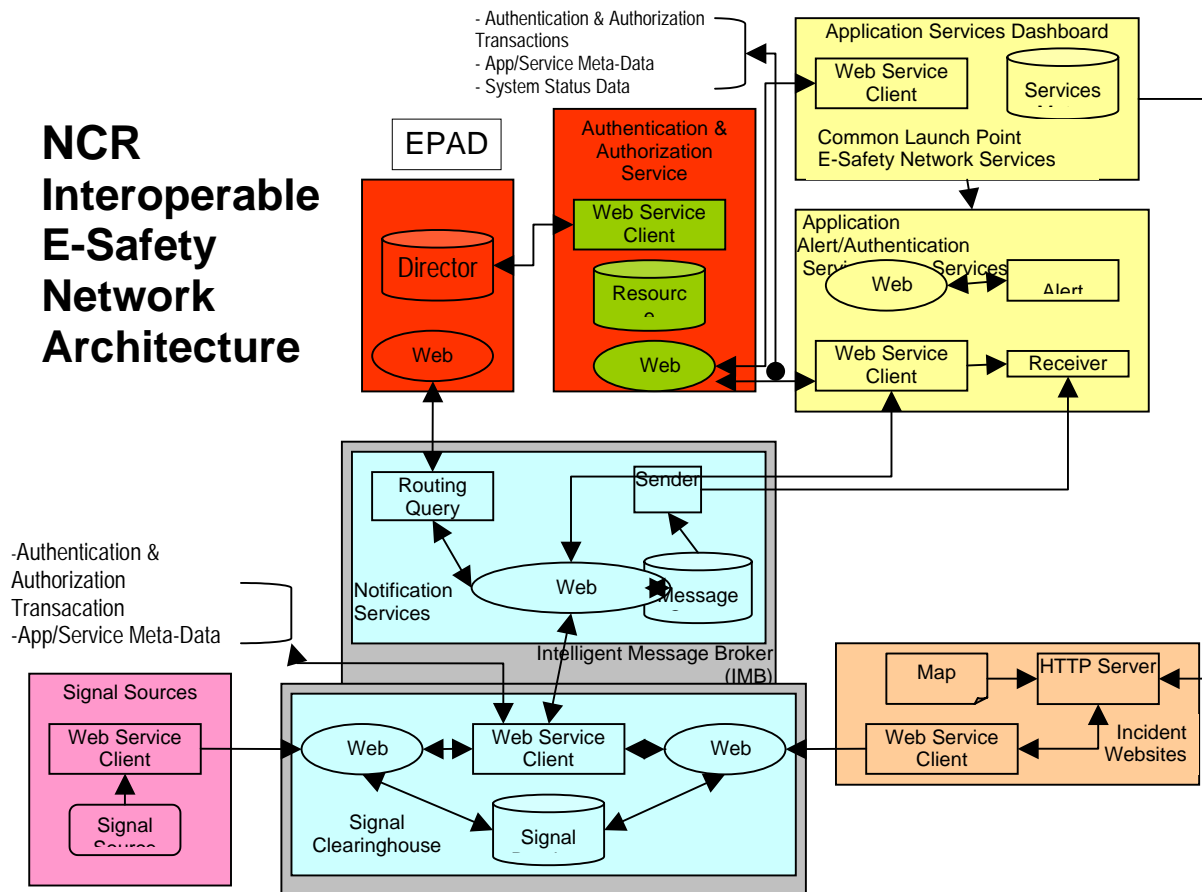


Figure 1: The E-Safety Network Architecture

2.1 EPAD

EPAD will be the authoritative source for emergency contact information. It is the core registry that will enable the other services such as dynamic notification of responder agencies in the event of an emergency signal. The EPAD database can be queried by various criteria, including geospatial data that describes the jurisdiction of agencies that have been previously registered. In this way an incoming emergency signal can be correlated to registered agencies. An agency registering with EPAD will be able to specify:

- Agency contact information, including:
 - One or more phone numbers,
 - One or more fax numbers,
 - One or more email addresses,
 - One or more electronic destinations for the delivery of messages,

- One or more emergency contacts, each allowing for:
 - Name
 - Designation
 - Phone Numbers
 - Fax Numbers
 - Email Addresses
 - Mobile Phone Numbers
 - Pager Numbers
- Agency jurisdiction, described geospatially,
- The Agency's area of interest for receiving emergency and incident information – for each incident type,
- The types of incident and emergency information the agency wishes to receive,
- Status of the Agency: Pending, Active
- Administrative and technical information:
 - Name
 - Designation
 - Phone Numbers
 - Fax Numbers
 - Email Addresses
 - Mobile Phone
 - Pager
- A user name and password for accessing EPAD.

This information is the foundation upon which the E-Safety network is built. Because of the requirement imposed by bulk loaders, which attempt to load contacts mechanically but may not have all of the information required to form a complete EPAD entry, the EPAD database will contain contacts that are still in the process of activation (pending approval or completion). Reports will assist in keeping the EPAD database clean and ensuring that there is follow-up even on entries that are bulk-loaded.

Contacts in the EPAD registry are always entered and managed through the Web Services interface, meaning that the point of entry is always the same. Web or platform-specific data entry screen can be developed against this Web Services backend.

2.2 Agency Alerts

Various agencies and interested parties can receive alerts from the E-Safety Network when emergency signals are generated within their areas of jurisdiction. In order to receive alerts, an agency must be registered with EPAD since the EPAD database is the basis for routing alert messages.

An agency can interact with the IMB in either of two ways, depending upon the model adopted for the E-Safety Network:

1. It can set up a SOAP server to receive incoming alerts sent directly from the E-Safety Network. When an alert is routed to the agency, the E-Safety Network connects to the server managed by the agency and transmits all pending messages.
2. It can use any desktop computer software that includes code to periodically poll the E-Safety Network for messages that have been stored for it. This polling may be assisted by the use of a secondary signal from the IMB so the agency computer knows when something new has been posted.

Once an agency receives its alerts, it can distribute them and process them in any number of ways in order to make them useful for its interested parties.

2.3 Signal Clearinghouse

The Signal Clearinghouse is the portion of the E-Safety Network most closely connected to the actual generation of emergency signals. It exposes a Web Service whose primary purpose is to receive incoming emergency signals. Those signals are then passed to the Notification Service modules in order to perform routing such that alerts reach the appropriate registered parties. The security of incoming signals is guarded by the use of SSL³ and HTTP header-based authentication to ensure that they are from legitimate parties.

Depending on the layout of the E-Safety Network, the Signal Clearinghouse may include logic to determine which parts of the network will handle notifications (distribute alerts) based on the signal. In the simplest model, there is no such logic and signals are always handed off to a well-known Notification Service server. In a more elaborate layout, the signal may traverse to different server depending upon the source or nature of the emergency.

In addition to forwarding signals for routing to agencies, the Signal Clearinghouse also stores signals in a dynamic database in order to preserve a “static image” of recently issued alerts. Another Web Service allows access to this database by any interested party (e.g. organizations using incident management websites or applications such as DMIS⁴).

2.4 Incident Websites

Emergency Management websites and applications can access a transient database of recent emergency signals through a Web Service provided by the Signal Clearinghouse. This makes it easy to plot current events on maps, or create other internal representation uses by those systems to track incidents or update status.

2.5 Intelligent Message Broker

A key feature of the E-Safety Network is its ability to accept emergency signals and route them to the appropriate parties based upon the EPAD directory. The implementation of this behavior has been dubbed the Intelligent Message Broker (IMB). The IMB consists of the combined functionality of the Signal Clearinghouse and Notification Service, using the incoming signal as the driver and the EPAD database as the source data by which routing is performed.

2.6 Signal Sources

Emergency signals can come from a variety of sources, including:

³ Secure Sockets Layer

⁴ Disaster Management Interoperability Services

- Sensors
- Vehicle Monitoring
- Operations Centers
- Public and private emergency agencies⁵

Emergency situations are communicated to the E-Safety Network electronically through the Web Service exposed by the Signal Clearinghouse.

2.7 Administration

Administration applications allow contacts to be entered into EPAD by efficient means. These applications may be in the form of GUI⁶ programs or web interfaces:

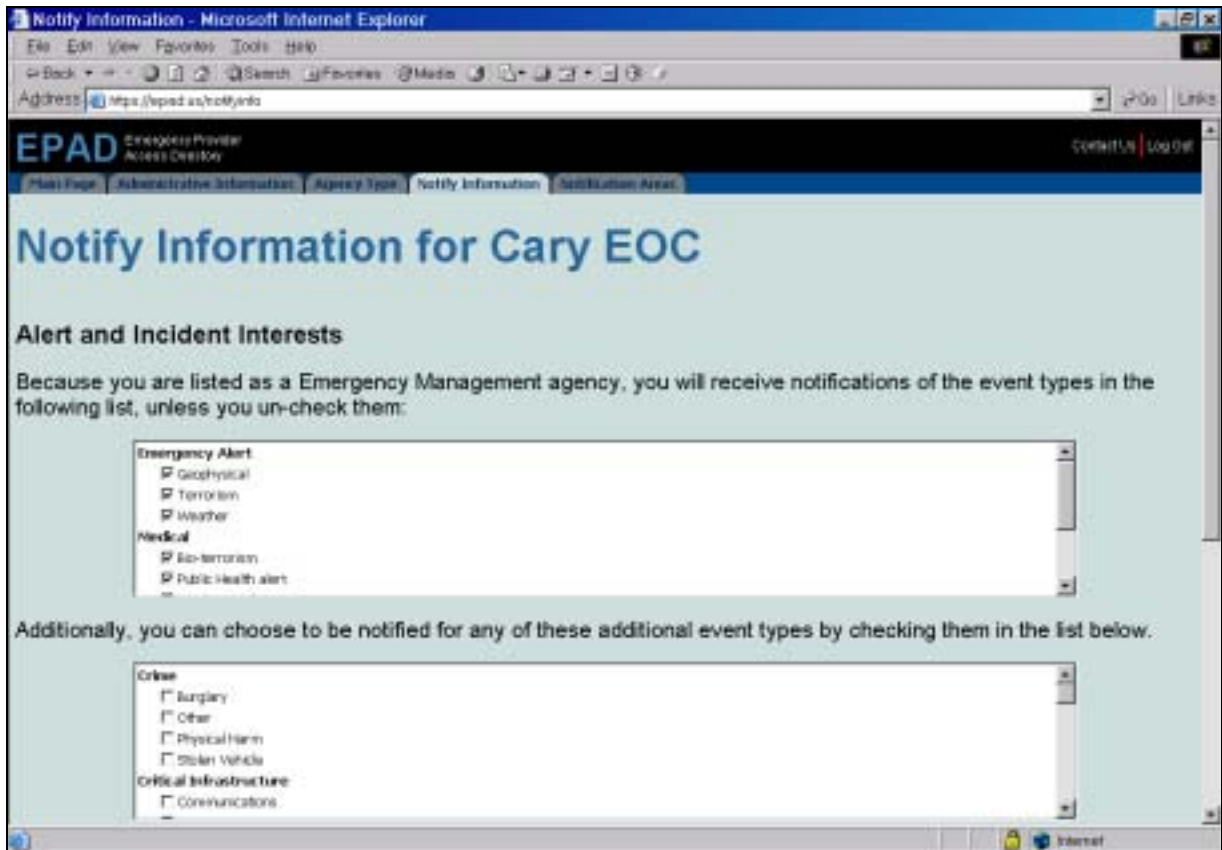


Figure 2: Example Administration Program for EPAD: Checking off incident notice types agency wishes to receive.

They may be also be implemented as bulk-loaders designed to replicate information from existing systems that already hold emergency contact information. In all cases, the information will be exchanged with EPAD through the Web Service exposed by EPAD.

EPAD must incorporate an intuitive way in which to define jurisdictional boundaries. This could be implemented by developing a GUI-based tool that displays map images and allows administrators to draw boundary lines at extremely fine levels of detail. To facilitate more accurate entry, EPAD may even draw upon city or county boundary information that is

⁵ EPAD is a directory of public and private emergency agencies – not of individuals. Agencies can choose to flow data from the E-Safety Network through to their staff or from them out into the Network.

⁶ Graphical User Interface, generally referring to elements that an operators sees on the screen

available from other official databases by leveraging the work of third party software vendors. Platforms (such as those provided by ESRI⁷) can be designed to interface with the EPAD administrative Web Service and offer integrated management to an established audience of operators, i.e. the ability of a user to upload their local boundaries into EPAD using a tool created by ESRI.

As a non-visual manager of agency information, the EPAD service must have a way to communicate with the outside world. This is the purpose of the EPAD Administration Web Service, which will offer functions through a SOAP/XML⁸ interface to programs that manage contact information.

The core of EPAD is the central management of contact information, and its Administration interface will reflect that functionality. As a result, the EPAD Web Service might expose operations such as the following:

- `addContact()` – Add a contact to the EPAD database
- `removeContact()` – Remove a contact from the EPAD database
- `updateContact()` – Update existing contact information in the EPAD database
- `getPartialContacts()` – Get a list of contacts that are missing information

This is not an exhaustive list of all operations that could be exposed by the EPAD service, but is intended as an illustration of the nature this interface. The information itself is exchanged in the form of XML data structures that are governed by an agreed-upon XML schema.

⁷ ESRI is an industry leader in geospatial and mapping software (www.esri.com)

⁸ SOAP stands for Simple Object Access Protocol, and it was developed by major players in the computing industry under the umbrella of the World Wide Web consortium. Its purpose is to offer a uniform, standardized way to access functionality available on servers by building upon the strengths of XML.

3 Intelligent Message Broker

3.1 Interfaces

The IMB will expose a variety of interfaces, each with a specific purpose relative to the overall function of the E-Safety Network.

3.2 Notification Service

The Notification Service is at the heart of the process for transforming emergency signals into alerts targeted at the appropriate agencies. The Notification Service connects with other entities on the E-Safety Network in three places:

1. By exposing a Web Service through which emergency signals are passed, typically after validation by the Signal Clearinghouse,
2. By invoking the EPAD Web Service to access information about agencies that may be interested in a given emergency signal, and
3. Interacting with agencies by exchanging alerts that have been deemed appropriate for those contacts. Various (and multiple) methods for transferring messages may be supported, but the simplest one is polling. The Notification Service holds alerts for each agency until they ask for them, which may be at a high frequency.

The Notification Service maintains message queues for the various agencies that are registered with EPAD. When an emergency signal is received, typically by the Signal Clearinghouse through the Web Service interface between the modules, a routing query is performed in order to determine which agencies need to be notified. Then an alert is placed in the message queue for each target agency. The alert is held in the queue until it can be transmitted to the agency.

Transmission of alerts to agencies can be performed in two general ways:

- Via a *push mechanism* in which the IMB opens a connection to a computer controlled by an agency and sends pending messages. This delivery pattern has natural responsiveness because any incoming signals will drive a process that immediately attempts to contact the appropriate agency computers. Those agency computers simply wait for an alert. The challenge of this approach is in the assumption that it makes that every agency will have a suitable server with a static address that can receive messages from the E-Safety Network. Setting up the appropriate hardware and software requires expertise, and incurs cost, that many agencies may not be able to easily afford.
- Via a *pull mechanism* in which the IMBs (or another server) passively holds queued messages until a computer controlled by the agency requests messages for the agency. In this arrangement, the “sender” on the IMB is actually another Web Service⁹ that is invoked by agency computers in order to retrieve pending alerts. Therefore, the agencies “poll” for messages on the E-Safety Network on a periodic basis. The frequency of those polls can be high enough that alerts are transmitted at intervals that

⁹ For handling polling, the IMB may offer other protocols such as HTTP GET in addition to SOAP/XML. These alternate mechanics are sometimes easier to program when only a particular subset of functionality is used.

approach real-time. The strength of this approach is its simplicity, and the fact that it offloads server complexities to the E-Safety Network rather than placing overhead onto the agencies. One drawback is that the E-Safety network must be able to service an extremely heavy load of “hits” since many agencies will be polling at the same time, and they may be performing frequent queries. This can be mitigated by state and regional servers.

The E-Safety Network may include “reference implementations” of code that might be used on the *push* or *pull* by an agency in order to lower the technical integration barriers to adoption.

There is also a hybrid approach in which a *pull mechanism* is primarily used, but is augmented by using an established secondary channel of some kind to inform an agency that it has pending alerts and needs to retrieve them. The advantage of this approach is that only a *pull* interface needs to be implemented, in addition to a signaling mechanism that is ideally very simple. The drawback is in finding a reliable signaling mechanism. One simple approach is for the IMB to contact the agency using an agreed-upon web URL, which tells the agency that it has pending messages. Implementing a web server is generally much easier for an agency than implementing a full Web Service, and a stock functional implementation can be developed fairly easily.

This hybrid approach is the *initial recommended approach* for the IMB, since it provides a simple single-interface message delivery model, but can also accommodate event-based delivery with only a little help from more sophisticated agencies. Further, the decision to implement the secondary signal can be made at the agency level, meaning that it serves all potential audiences.

3.2.1 Alert Push Services

As previously noted, the IMB Notification Service could support two models for the transmission of alerts to agencies. The first method is the *push* method, in which the IMB contacts computers operating within the control of the given agencies, and then transmits pending messages. The interface for this is an *outgoing* interface since the IMB will be contacting a Web Service running on an outside computer in order to transmit alerts. The alerts themselves are transmitted in an XML format¹⁰. The alert format for the *push* interface is the same as for the *pull* interface. The only difference is in who initiates the contact.

Push services assume that an agency is maintaining a Web Service that can respond to requests from the IMB. This Web Service must conform to the formats and functions expected by the IMB. In order to simplify the deployment of this model, and to lower the barrier to using it, a *reference implementation*¹¹ can be developed that offers drop-in functionality for agencies. For example, it may automatically comply with the needs of the IMB, and have an “out of the box” ability to receive alerts and store them in a simple folder or database.

¹⁰ The alert format may conform to the EPAD XML Signaling Format (EXSF) described in this document.

¹¹ A *reference implementation* would be developed by ComCARE, with the purpose of either illustrating how to handle incoming alerts from the IMB, offering standard out-of-the-box functionality, or both.

3.2.2 Alert Query

The second method for the transmission of alerts to agencies is the *pull* method, in which the IMB waits for requests from computers operated by registered agencies to retrieve pending messages. The interface for this is an *incoming* interface.

The IMB Notification Service sets up a Web Service that can be accessed via SOAP/HTTP. Alternately, and perhaps in parallel, it may offer secondary interfaces such a HTTP/GET that may be easier for some agencies to leverage. In all cases, the alerts themselves are returned in an XML format, perhaps using the EPAD XML Signaling Format (EXSF) described elsewhere in this document. The alert format for the *pull* interface is the same as for the *push* interface. The only difference is in who initiates the transfer.

3.2.3 Signal Notification Handoff

This interface is the mechanism that links the Signal Clearinghouse with the Notification Service offered by the IMB. This service can also be invoked directly by an administrative application that allows alerts to be sent manually.

3.2.4 Signal Source

Signal sources are the start of the alerting chain. They produce messages based on emergency events, and send them to the IMB using the Signal Clearinghouse Web Service. The format of these signals follows the Common Alerting Protocol (CAP) or other formats that can be interpreted by the IMB. The actual message may be wrapped in a common envelope format that allows new types of messages to be handled with some uniformity against the existing ones.

3.2.5 Signal Query

Incident management websites and services can query the IMB to get updates on the current state of signaled events. This information is stored by the Clearinghouse as signals arrive, and provides an easy source for emergency software to plot current events on maps (as well as other formats useful in that context).

3.3 Sender Control

The wrapper used by the IMB for representing signals must allow for explicit filtering of the target agencies governed by the sender. These optional criteria will affect the way that the Notification Service queues alerts based upon that signal. Signals that are sent manually through the EPAD administrative interface can then be targeted more precisely at specific agencies or jurisdictions. Signals that are sent mechanically can also supply criteria, but are more likely to rely on the default delivery behavior assumed by the IMB.

For allowing sender control at the signal source level, the E-Safety Network may consider implementing a *sender registry* in addition to the registry that constitutes EPAD. This new registry could retain information on various signal sources and automatically apply proper filters when signals are generated and entered into the network. This would allow simple filtering to be applied even to automated signals, such as Automatic Crash Notification messages. The only definite requirement for the signal source is that it identify itself so that its registration and settings can be honored.

3.4 Receiver Control

There are two ways in which agencies can filter alerts based on their own interests:

1. Declare areas of interest when registering the agency with EPAD. In this way, the IMB will only send alerts when they are of interest to the given agency. The described areas of interest are treated much like other criteria when the Notification Service distribution logic operates on a signal.
2. Accept all alerts and simply ignore ones that are irrelevant. This may involve downloading all alerts and filtering them client-side, or handing criteria to the IMB when retrieving alerts that serve to restrict the subset being returned.

The first approach preserves bandwidth by preventing the transmission of unwanted alerts, while the second option allows an agency to handle its areas of interest in a dynamic fashion.

4 Network Distribution

Because it is the easiest to illustrate, the model considered so far has assumed a single, centralized implementation of EPAD and the IMB. It is possible, however, for alternate distribution models to be developed in order to satisfy a wide variety of deployment needs. Each distribution model has its own characteristics, some of which are discussed here, but the goals for a distributed EPAD include:

- Load balancing,
- Maximizing local control, and
- More directly modeling the distribution and needs of stakeholders

It should be noted that, with only a few exceptional scenarios, the ability to support multiple agency databases will affect the design of the entire E-Safety Network and not just the EPAD design. This is because the EPAD database is the core information reference used by the IMB. It must be clear to the Notification Service which EPAD database is to be used for routing alerts based on the given signal. Most approaches can be accommodated, but the distribution requirements must be considered throughout the design so that the flexibility exists for the IMB to leverage more dynamic deployment models.

It is possible that the E-Safety Network can support one or more of these distribution models simultaneously, if that is deemed to be a useful requirement. If the deployment requirements are not immediately clear, there is virtue in adopting a simple model initially since a straightforward approach will be easier to design, develop, and optimize. In many cases a more sophisticated model can be implemented in phases. The key is in achieving the best balance of priorities.

4.1 Independent, Parallel EPADs

Different regions may decide to implement their own self-contained EPAD services, perhaps to maximize control over the content or for some other logistical or legal reason. In this case, each region would have an independent EPAD that is maintained independently. The following image depicts two independent EPADs symbolically:

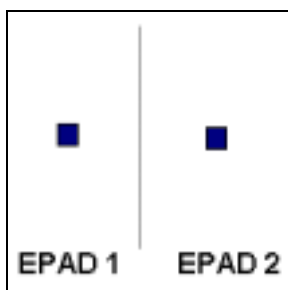


Figure 3: EPADs Existing Side by Side

Such a distribution can have the following characteristics:

- The maintenance/administration applications can be the same software used by the central EPAD (or other regions).

- If a distinct IMB is also maintained separately, as a mirror image of the entire E-Safety Network, then all function the same except that the agency database is isolated.
- Without special consideration, such as import end export functionality, there is no sharing between EPADs. Each is for private (relative to the region) use only.

These are the qualities that a simple parallel distribution of EPAD would exhibit.

4.2 Central EPAD with Linked Children

This relatively simple distribution model has a central EPAD instance and supports localized EPAD instances that connect directly to the central one. This offloads certain contact management functions to the local instance, but at the same time allows the central EPAD to easily operate as a single unit when required.

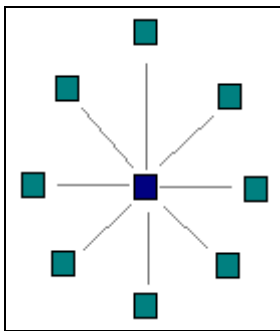


Figure 4: Central EPAD with Children

The characteristics of this approach include:

- Agency maintenance/administration is performed against the local EPAD instance. The maintenance/administration applications can be the same software used by the central EPAD (or other regions). All validated changes to EPAD are “rolled up” into the central instance. Because all information is available centrally, the IMB can function against the main EPAD when distributing alerts to agencies.
- When routing alerts to agencies, the IMB always uses the central EPAD instance which contains the conglomerate of all locally published agency information. There is distribution in the registration and maintenance functions, but the routing of alerts remains centralized.
- Local EPAD instances provide a modest level of autonomy and privacy.

This model can be enhanced further by also distributing the IMB instance, so that localized EPADs can also service localized alerts. This provides more of the benefits of distribution, such as scalability, than simply creating a network of EPAD instances that must always meet in the middle.

4.3 Independent, Interlinked EPADs

This deployment scenario allows for a network of EPADs that relate to one another in arbitrary ways. There is no predefined or imposed structure, at least technically, and there may be more than one separate or interlinked network.

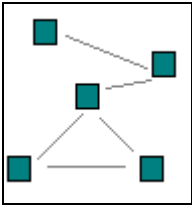


Figure 5: Freeform Networked EPADs

It has the following qualities:

- The entire E-Safety Network, including both EPAD and the IMB, are implemented at each node. This makes each instance a fully functioning alert network.
- Each instance maintains a database of other instances that *it trusts* with its own agency information. Signals posted to a node can traverse through the network (and thereby generate alerts) as allowed by the defined trusts and the entry point of the signal.

This layout is the most technically flexible one; it is also the most complex unless some constraining assumptions are applied. In practice it may be useful to establish “supported network patterns” in order to provide a more orderly experience.

4.4 Interlinked EPAD Hierarchy

An EPAD hierarchy offers a good balance of flexibility and control. It is also relatively easy to visualize, simplifying matters for developers, administrators, and stakeholders.

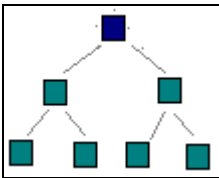


Figure 6: EPAD Hierarchy

There are several possible ways to implement a hierarchy, each with slightly different semantics regarding who “owns” the data. Two of these approaches are:

- **Bottom Up.** Most contact information is collected at lower levels through independently maintained EPAD instances. Registration and update activities are “rolled up” into the *parent instance*, successively, until they reach the central EPAD instance. Because the central instance has a superset of all usable agency information and E-Safety Network to operate on it, the individual nodes do not need to implement an entire IMB but rather only the EPAD. This presents a relatively low barrier to operating a node in the hierarchy. For organizations that do not wish to (or cannot) operate an EPAD instance, they may still operate directly against the central instance.
- **Trickle Down.** It may not be desirable for all of the agency information to be rolled up to the central level. Some organizations may wish to control their contacts more tightly. To accommodate this situation, the processing in the hierarchy can be *inverted* so that the central database no longer holds a master superset of agency information. Rather, limited of data is held centrally with major portions housed

within the tree. When a signal is received and the IMB is performing its routing query, it attempts to service the jurisdiction from its own repository. If the information is not available, the routing query is handed to the next child node in the hierarchy until matches are found.

In short, the *Bottom Up* approach provides better alert-time performance but requires organizations to readily hand over contact information for the central repository. The *Trickle Down* strategy allows finer control of information by EPAD instances, but at runtime the process of routing alerts becomes more computationally intensive.

4.5 Hybrid Approaches

The distribution models discussed here are only a few of the possible frameworks. In all likelihood, the “perfect” model will combine elements of one or more of these approaches in order to balance all requirements and constraints sensibly. ComCARE’s expectation is that there will be multiple IMBs.

5 Security Considerations

5.1 Authentication and Network Security

A fundamental element of security is the assurance that those who are accessing the E-Safety Network are authorized to access it. This assurance is granted by the process of *authentication*, a significant portion of which is built into the Web Services model.

5.1.1 SOAP/HTTP Authentication

Web Services rely heavily upon the SOAP protocol, and the SOAP protocol rides upon HTTP as a transport method. HTTP is the same language used by the World Wide Web to serve HTML web pages. As a result of the synergy between SOAP and the World Wide Web, Web Services have inherited many security features and limitations from the internet.

5.1.2 Validating Identities

The process of identifying a Web Services client generally involves transmitting a login name and password combination to the IMB. With basic HTTP, these credentials are embedded in every request as a part of the HTTP header. The absence of credentials where they are required causes the WWW server to send back an error code (401 Unauthorized) that normally causes the browser to present a login screen and resubmit the request with the proper values.

Since Web Services operate over HTTP, they can also use basic HTTP authentication by supplying credentials (issued by EPAD during agency registration) in the headers for each request. One problem with this approach is that, with basic HTTP authentication, passwords are sent as plain text, making them vulnerable to interception. The IMB can also choose to implement its own variations on this theme. For example, it may employ token-based authentication. Token-based authentication requires users to supply credentials through a dedicated secure channel. The SOAP server then responds with an *authentication token* which can be used for all subsequent requests. This reduces the risk of credentials being compromised because they are only sent once, and then subsequently represented by a secret value used only for the duration of that session.

5.1.3 Protecting Data in Transit

Once the identity of a client has been validated, the IMB and the agency (or signal source) will begin exchanging information in the form of SOAP/XML messages. The contents of the conversation¹², the messages, must be secured so that electronic eavesdroppers cannot easily intercept them. In many cases, the process of authentication does not inherently protect from this, making it possible for an attacker to read or alter the conversation even though identities were validated at the instant the connection was made.

Securing the conversation for its duration can be accomplished though the use of SSL or VPN¹³, which applies encryption or hashing to everything that travels across the connection –

¹² In this context, a “conversation” is any logically related series of messages between entities on the network. Typically, anything sent or received over the course of a single validated connection. The authentication process ensures the identity of the parties as the conversation is initiated, but without further protection it is possible for information theft to occur after authentication if data is intercepted by a third party while in transit.

¹³ Virtual Private Network

including any header information. In the case of VPN, the solution may incorporate its own authentication scheme that frees the Web Service from implementing it. The downside is that a VPN usually requires its own software running at each end of the connection. In addition, creating a scalable VPN-based network is not a trivial task.

The E-Safety Network can implement encryption by exposing its Web Services through SSL. SSL offers the same kind of privacy offered by secure web sites. Private keys used by SSL will be stored on the IMB servers. They are extremely sensitive and must be secured not only by technology, but by self-checking procedures as well.

5.2 Roles and Privileges

Building upon strongly defined network security, EPAD will also have its own system of roles and privileges. Whereas authentication is generally concerned with *who* has access, roles and privileges are more focused on *what functions* individuals are allowed to exercise.

Some of roles that might be defined include:

- Administrators
- Alert Senders (defined by type of alert)
- Users

Each role has corresponding privileges supporting that role. Individuals may function in one or more different roles, meaning they can perform any function assigned to any of the functional groups to which they belong.

All roles are implemented on the server side, protected by the “firewall” of the Web Services interface. As opposed to implementing rules in client software, this server-side enforcement ensures that rules cannot be circumvented because the service (including security) is a “black box” to its users.

In a distributed model, IMBs may employ their own particular security features associated with sharing restricted information. In such cases, roles and privileges defined behind the EPAD interface will also need to be compatible with those of the IMB whenever restricted information is part of a message payload. EPAD will need to recognize and pass through appropriate authentication and role-based security information to its IMB connections. The degree of coupling imposed by this requirement may have significant influence on the choice of implementation mode defined in Section 5 above.

6 Data Storage

6.1 EPAD Entry Characteristics

EPAD will store contact information for registered agencies. This information may consist of multiple address and contact names and numbers, as outlined previously. This data can be readily stored in a relational database such as Oracle or SQL Server. Proper normalization and constraint definitions will help to ensure data integrity, and well-planned indexing can keep performance sufficiently high. Information on jurisdiction boundaries must be stored as well. When entering boundary information, the administrator is likely to see the jurisdiction as it is laid out on a GIS map, as shown below:

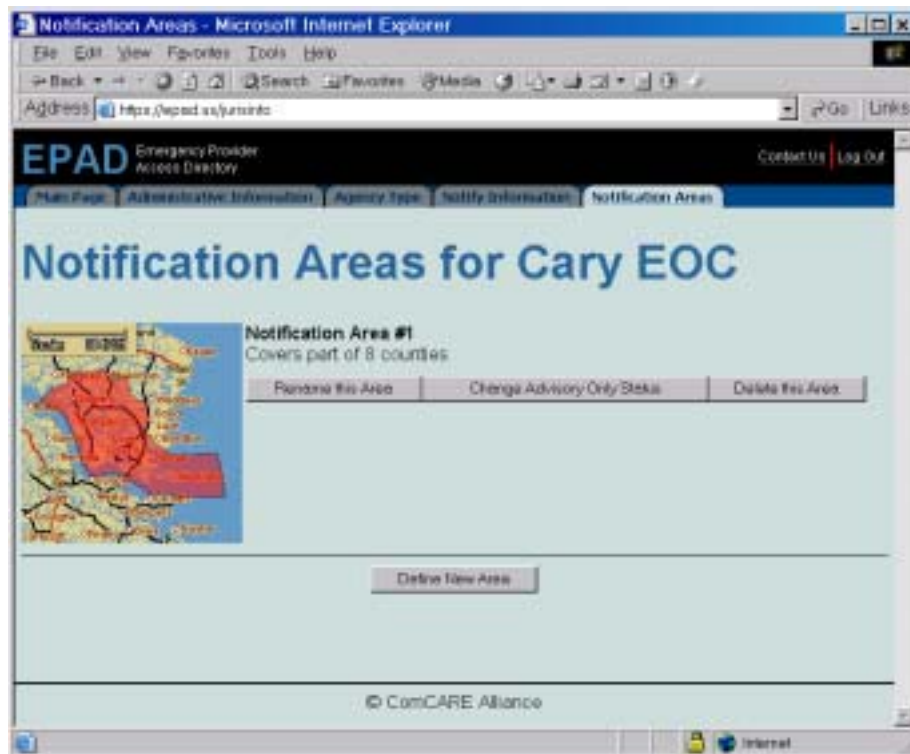


Figure 7: Defining Jurisdictional Boundaries

Internally, this information must be stored with consideration for how intersection computations will be performed during routing queries. The bounded area (or perhaps multiple areas) will likely be stored as a set of geo-encoded points that comprise polygons:

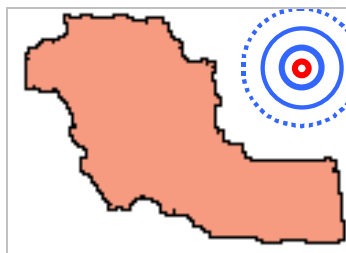


Figure 8: Geocoded Jurisdictional Polygon

Even though this point data is mathematical rather than Geospatial, which is generally known to consume nontrivial amounts of network and storage bandwidth, the polygon data can still be rather intensive. The number of points required to describe a jurisdiction may be substantial, and the EPAD database must be designed from the beginning not only to accommodate this data, but to search it efficiently. This is particularly true because users have made it clear they want different “interest boundaries” for different types of incidents.¹⁴ This may require boundary data to be stored in a different repository than the main contact information for an agency, one that is specialized for handling this type of information.

6.2 Performance Issues

The boundaries of jurisdictions can be represented internally in a number of ways, but in every case it will be fairly data-intensive. When an emergency signal is sent, the Notification Service will query EPAD to determine which agencies need to be notified based upon a variety of matching characteristics (including geography). This geographic hit-test will be intensive relative to the other possible criteria. Some strategies that EPAD might choose to employ in order to reduce the impact include:

1. In multi-criteria queries, always perform the least expensive tests first. In this way, agencies that do not meet simple criteria (such as the type of agency) do not warrant further matching based on geography.
2. Organize geographic data for quick lookups. This may mean spending more time preprocessing geo-encoded polygon data, when agency data is created or changed, so that the data is organized for fast comparison. Always organize data to optimize the operation that is the most frequent or critical.
3. Implement strategies, such as bounding box comparisons, to enable quick determination when an event is definitely *not* within a given jurisdiction. Reserve the extended, more detailed boundary intersection computation for events that are close to a jurisdiction.

The handling of boundary information is perhaps one of the most challenging problems for EPAD. In order to make sure that EPAD performs as well as possible, this area in particular must be designed carefully.

6.3 Import from Existing Sources

One powerful way to immediately bolster the relevance of EPAD is to import contact data from existing systems that already contain widely-accepted information (such as from FRED¹⁵). Bulk loaders can be designed to record contacts through the same interface used by the GUI to populate entries manually. The basic process is:

1. Decide on a valuable source of contact information,
2. Write a bulk-loader that reads contacts from the system and submits them to EPAD through the defined Web Service interface, and
3. Manually clean up information that cannot be translated mechanically.

Because bulk loaders attempt to load contacts mechanically, but usually will not have all of the information required to form a complete EPAD entry, the EPAD database will contain contacts that are still in the process of activation (pending approval or completion). Bulk loading also presents a higher risk of duplicate or conflicting entries. EPAD will rely strongly upon reports to assist in keeping the EPAD database clean and ensuring that there is follow-up for entries that are bulk-loaded.

¹⁴ Indeed, jurisdictional boundaries for a single agency can vary by incident type (e.g. fire v. hazmat).

¹⁵ Facility Resource Emergency Database

7 Scalability Strategies

7.1 Assessment of scalability challenges

The IMB may encounter a large volume of concurrent emergency signals, proportional to the distribution of EPAD usage across the country and the current alert status of the nation. Each emergency signal represents one or more hits to the Signal Clearinghouse Web Service, that are at the front end of an intensive process that routes alerts to registered agencies. While emergency signals are received, the Signal Clearinghouse may also receive requests from external incident management services for a snapshot of the current signal set. If the E-Safety Network is widely leveraged, then the Signal Clearinghouse will require scalability at potentially impressive scales.

Each incoming signal results in a call to the Notification Service in order to determine which registered agencies must receive alerts. The process of matching signals to agencies includes data-intensive geographic matching that is performed (at a minimum) for any jurisdiction near the emergency area. Alerts routed to specific agencies must be written to message queues until they can be delivered to those agencies by a supported transmission method. At any given time, there may be a substantial number of queued messages, and the IMB must ensure that there is minimal risk of message loss.

Agencies will receive alerts from the queue maintained for it by the IMB Notification Service. For agencies operating in *push* mode, the Notification Service must contact servers on an outside network managed by those organizations. Although the supported transfer protocols will be dictated by the E-Safety Network, the process for initiating those transfers must be designed with scalability in mind. For agencies operating in *pull* mode, their computers will initiate contact by invoking a Web Service exposed by IMB Notification Service. The frequency of queries is controlled by the initiator (the agency), and in many cases may be intended to approximate real time. Given a widespread leverage of EPAD, the scalability requirements for this polling service could be massive.

The core EPAD database will be relatively static, though the volume of updates to the core agency contact information will be relative to the number of registered agencies. The EPAD Web Service for managing agency information must be scalable to the projected scope of deployment. By far, EPAD will receive most of its activity in the form of queries from the IMB attempting to routing alerts to relevant agencies. This EPAD query service will house the matching algorithms that filter based on types and jurisdictions.

The entire system of interconnecting pieces that comprise the E-Safety Network must be engineered in such a way that each piece can be rescaled without requiring significant reengineering of the other parts.

The Web Services software comprising the IMB, on one end of the problem, is ultimately concerned with handling enough “hits” to support the goals of the project. On the other end, emergency signalers are concerned that alerts reach the correct party in enough time to make a difference. In order to produce meaning scalability requirements, it is crucial to determine where those interests intersect.

7.2 High Level Technical Approach

EPAD will use a Web Services framework (which services SOAP requests) that supports the clustering of servers, along with a design that can leverage this capability. The intent is to create a situation where demand for increased scalability can readily be addressed by adding new servers if necessary. Additional servers would then allow the network to handle more requests in parallel. A solid, cluster-centric design is not trivial to create, but many modern tools (such as application servers) are geared toward this, and a system designed around such an approach will likely demonstrate longevity.

In a distributed deployment of EPAD, it is also crucial to consider the impact of replication and carefully consider how changes will be propagated across the network of EPAD instances. The communication between instances can itself require significant bandwidth or processing power. More centralized deployment models have fewer complications, but offer less control over information by participating agencies.

8 Data Integrity

The E-Safety Network will service a high volume of requests on multiple fronts. It is crucial to protect the integrity of the information that it serves, and this requires a multifaceted approach.

8.1 Data Revalidation and Housekeeping

In addition to validating EPAD entries as they are activated, there must also be a proactive process in place for *revalidating* existing information on a periodic basis. Various alternatives are now being explored. EPAD might employ an expiration process in which certain entries are marked as stale unless reviewed and revalidated periodically by agency personnel. When routing alerts, the E-Safety Network may choose to ignore agencies whose revalidation is seriously overdue. This will ensure that EPAD acts only on information that is relatively fresh.

Reports are an important part of the validation and revalidation process because they can highlight areas that require attention and would otherwise not receive it. The agency administrators should receive a notification automatically when revalidations will be required.

Validation and housekeeping of the EPAD will require dedicated processes that run on a regular basis to scan the entire database for specific anomalies. When issues are identified, the appropriate action, such as generating administrative messages or flagging entries, must be triggered. The appropriate actions to be taken must be defined prior to designing these housekeeping routines.

8.2 Administrative Messages

Agency administrators must be notified when some condition related to their agency requires attention. These notifications can take the form of screen messages that appear while working in the EPAD administrative program, or in more extreme cases they may be delivered to the agency as alerts. The appropriate means of notification must be considered.

8.3 Database Design

The linchpin of data integrity is the proper application of database techniques, because this ensures validation at a very fundamental level. It starts with a solid, normalized database design that minimizes the redundant data. It is cemented with the application of constraints and indexes that serve to ensure that all records are consistent with other related records.

8.4 Message Queues

The IMB Notification Service will generate alerts and transfer them to agencies. In between the time they are generated, but before they can be transferred to the intended agencies, they are held in a message queue. This queuing approach improves data flow by allowing the generation of alerts and the transfer of alerts to be decoupled.

Message queues must be designed for lossless, nonvolatile operation in the event of a system failure. That is, in the event that the IMB goes offline, any alerts that have been generated but not yet delivered must not be lost. As soon as the IMB comes back online, the contents of the message queue should be the same so that agencies can continue to receive all alerts reliably. This will make the Notification Service more resilient to brief service interruptions.

8.5 Backup and Recovery

A critical element of data integrity is proper backup procedures. Backups serve as a source for disaster recovery in the event that the primary EPAD database experiences profound corruption. Copies made on a regular basis and kept offsite offer a form of insurance against this - though the restoration procedure should be tested periodically to ensure that a restore process will actually work.

8.6 Auditing

Changes that are made in the EPAD database should include a signature of when the record was last updated and by whom. For a more complete history, a full audit trail can be generated on all changes that are made. This level of auditing must be built into the design of the network from the outset, since it can be more difficult to track this level of detail later.

Auditing can also carry performance penalties that must be considered and balanced with the level of information that is actually useful.

9 Reporting

Over time, any contact database can become stale or infested with duplicate entries that make it hard to determine which contacts are truly valid. Using an application in situations where lives are continually at stake means this is an issue that must be addressed. The nature of EPAD requires that it have the capability to produce reports based on various attributes of the agency entries it manages. These reports will serve many purposes, including:

- Helping to keep the database clean by allowing agency administrators to easily find elements that need attention,
- Supporting bulk registration (or multi-step registration) processes by flagging accounts that are missing crucial information,
- Lowering the barrier to participation in the E-Safety Network by simplifying maintenance of EPAD entries by agencies,
- Maintaining the accuracy of EPAD and bolstering its value in a critical area of need.

Some of the reports that might be useful include:

- Statistics regarding the activity levels of various agencies and signal sources,
- Statistics regarding activity levels based upon locality,
- Lists of agencies whose contact information has not been revalidated for a specified period of time,
- List of contacts that are incomplete and require more information prior to activation,
- GIS overlaps of jurisdictional claims by the same profession (e.g. two sheriffs claiming the same part of a county)
- GIS “holes” in registration by professional area, to be referred to higher state authorities.

Many of these reports can be generated manually by the appropriate participants (e.g., agency administrators), although some may be generated and send out based on an automated schedule.

Since generating reports is generally computationally intensive and often based on information from audit trails created during real-time activity, the database design must take into account the kinds of reports that are to be produced. This will ensure that there is enough data to generate the desired set of reporting tools.

10 Web Services Overview

For the E-Safety Network and EPAD, Web Services provide the perfect foundation on which to build a standards-based directory service. A Web Service is an application component that can be described, published, located, and invoked over a network using standardized XML messaging. Defined by new technologies like SOAP, Web Services Description Language (WSDL), and Universal Discovery, Description and Integration (UDDI), this is a new model for creating e-business applications from reusable software modules that are accessed on the World Wide Web.

Stated simply, they are an increasingly popular, standardized way of integrating disparate systems and applications. Essentially, Web Services is a Service-Oriented Architecture built on the SOAP standard, and owes much to older distributed computing protocols like RPC, RMI and CORBA¹⁶.

10.1 Service-Oriented Architecture

A service-oriented architecture is one that has a robust service layer. The services in the service layer have the ability to be invoked over a network. The technologies used to invoke the interfaces exposed by the services stress interoperability. The services in the service layer also stress location transparency so they may be discovered and used dynamically.

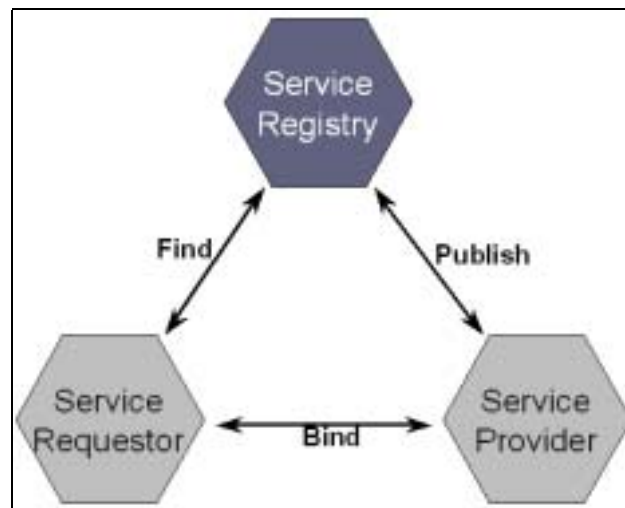


Figure 9: Service-Oriented Architecture

Typically, clients will know the Web Services they need, and these will reside on a well-known server located on their network (or on the internet) and are addressed using a URL.

¹⁶ RPC stands for Remote Procedure Call, and refers to a well-understood paradigm of invoking functions residing on one machine from a remote machine. Remote Method Invocation (RMI) is an object-centric implementation of RPC for Java which is leveraged by the Java Enterprise Edition (J2EE) computing platform. Common Object Request Broker (CORBA) also used RPC idioms to offer remote access to objects residing on another machine. Like these protocols developed before it, SOAP is designed to allow controlled access to functionality over a network except it is built upon XML.

10.2 SOAP

10.2.1 Overview

For several years, CORBA and DCOM¹⁷ have been the two major competing standards for exposing functionality in a standard fashion. The explosion of Web Services, however, eventually resulted in a vacuum of platform independent standards that fit well within the stateless model of the Internet. The protocol that was developed to fill this void is SOAP. From the SOAP 1.1 Specification:

SOAP is a lightweight protocol for exchange of information in a decentralized, distributed environment. It is an XML based protocol that consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules for expressing instances of application-defined datatypes, and a convention for representing remote procedure calls and responses.

Described simply, SOAP is a messaging protocol that relies on the widely adopted XML and HTTP standards to allow Web Services to be established in an open manner. It can be used to implement RPC-like functionality, in which case a SOAP request is a semantic rough equivalent to a method call in an object oriented language. It can also be used to implement fire-and-forget messaging patterns.

An open specification, SOAP is mostly neutral to any particular programming model, operating system, or environment. SOAP does not attempt to change the way a system works, but rather to offer a way to *wrap* its functionality so that it can operate in a distributed manner, to the degree that is appropriate for that system.

11 Sample EPAD Implementation

11.1 EPAD Architecture

Design of a Web Service starts by describing the operations that the service will provide to the user community. Those operations may be a totally new facility previously unavailable, or they may be wrappers designed to standardize access to existing functionality. In all cases, the Web Services standards serve to hide the implementation details and insulate all parties from future changes.

11.2 API Design

11.2.2 Operations

The core of EPAD will be the central management of contact information. The EPAD service must expose operations to facilitate this and offer the functionality that will be needed by responder-related organizations. The EPAD Web Service might expose operations such as the following:

¹⁷ Distributed Component Object Model (DCOM) is an object-centric RPC model integrated into the Windows environment that competes at many levels with CORBA.

- addUser() – Adds a contact to the EPAD database
- removeUser() – Removes a contact from the EPAD database
- updateUser() – Updates existing contact information in the EPAD database
- query() – Locates contacts currently in the EPAD database

While this is not an exhaustive list of all possible operations that can be offered by the EPAD service, this list is a likely a subset of the operations to be exposed and is used here for the purpose of discussion. It is quite possible that other administrative functions, perhaps dealing with notifications or other extended functions will be required.

11.3 Data Structures

11.3.1 Documentation of Data Structures

Consumers of the EPAD service receive information on the various data structures by consulting the WSDL for the service. WSDL describes the EPAD service and all required structures in enough detail that machine-generated wrappers can be produced for various languages and environment. For example, the Apache AXIS tool suite can read WSDL and produce immediately-usable Java classes for accessing the service being described. Other environments such as .NET offer a similar ability to interpret these descriptors. All they need to know is where to find the WSDL file for EPAD.

While WSDL is often sufficient for generating machine-level interfaces to access EPAD, it is often useful to also provide human-readable documentation to describe best practices that accompany those elements. That is, the machine conventions are most powerful when they are aligned with the proper use and intent of the information being managed.

10.2.2 Sample EPAD Data Structures

Person
firstName: String lastName: String middleName: String prefix: String suffix: String

JurisdictionType
value: String { “Unknown”, “Agency Approval Authority”, “Department of Transportation”, “Emergency Management”, “EMS”, “Fire and Rescue”, “Hospital”, “Law Enforcement”, “Municipal/Supervisory”, “Primary PSAP”, “Private Sector Event Reporter”, “Public Health”,

“Secondary PSAP” }

JurisdictionLocality

value: String { “Unknown”,
“Municipal”,
“County”,
“State”,
“Federal” }

InterestLevel

value: String { “Unknown”,
“Advisory Only”,
“All” }

Severity

value: String { “Extreme”,
“Severe”,
“Moderate”,
“Minor”,
“Unknown” }

Criteria

locations: *array of* Area
severity: Severity
interestLevel: InterestLevel

Organization

name: String
jurisdictionLocality: JurisdictionLocalityType
jurisdictionType: JurisdictionTypeType
url: String

User

eMail: String
address: Address
criteria: Criteria
destinations: *array of* Destination
fax: String
homePhone: String
mobilePhone: String
name: Person
workPhone: String
organization: Organization

Point

points: *array of* Point

Area

description: String

PointPoly (extends Area)
lat: double long: double
FipsPoly (extends Area)
fips: String
Destination
destinationData: String

11.4 Stub Generation

There are many ways to invoke a remote Web Service. HTTP is used as the transport, so anything capable of sending an HTTP packet is a candidate. Programmatically, it ranges from building a SOAP call with a low-level API, to static invocation using a pre-generated stub, to dynamic invocation, which generates the stub at the time of the method invocation. Most common languages have toolkits to generate static stubs, and some tools have dynamic invocation abilities.

11.5 Authentication

Authentication is handled through basic HTTP. An “Authorization: Basic” header must accompany *every* request. Typically, the stub implementation will provide a convenient way to set it.

Java Example

“epad” is the stub class in this example:

```
((EpadSoapBindingStub)tie).setUsername("username");
((EpadSoapBindingStub)tie).setPassword("password");
```

VB Example

“ws” is the stub class in this example:

```
Dim myCred = New System.Net.NetworkCredential("username", "password")

Dim myCache = New System.Net.CredentialCache()
myCache.Add(New Uri(ws.Url), "Basic", myCred)
ws.Credentials = myCache
ws.PreAuthenticate = True
```