Detailed Design of the systems

for

Electronic Data Exchange between Police Agencies and Prosecutor Offices in Bosnia and Herzegovina

Ver.2.2.1

Sarajevo – November, 2010

NOTE:

AFTER SINGING THIS DOCUMENT ALL PREVIOUSLY SIGNED "DETAILED DESIGN DOCUMENTS" WILL BECOME OBSOLETE.

AUTHOR:

DONE BY MINISTRY OF SECURITY WORKING **GROUP (ACT-DECISION NUMBER 04-04-3-6285-22/09 FROM 23 NOVEMBER 2009)**

Contents

1. INTRODUCTION 5

1.1 ACRONYMS 5
1.2 OBJECTIVE 7
1.3 EXCLUSIONS 7
1.4 INTENDED USERS 7
1.5 ARCHITECTURAL GOALS AND CONSTRAINTS 7
1.6 GUIDING PRINCIPLES 8
1.6.1 Scalable 8
1.6.2 Flexible 8
1.6.3 Standards-Based 8
1.6.4 Reliable 8
1.7 WEB SERVICES DEFINITION 9

2. SOA ARCHITECTURE AND TOPOLOGY 9

2.1 COMPONENTS 11
2.2 OPERATIONS 13
2.3 WEB SERVICES STACKS 13
2.3.1 Wire "Stack" 13
2.3.2 Discovery Agencies "Stack" 16
2.3.3 Service Bindings 20
2.4 ENCODING 21

3. LOGICAL DESIGN 23

3.1 CAPACITY 24
3.1.1 Central Point Service (Enterprise Service Bus and WS Repository) 24
3.1.2 Elementary Web Services 24
3.2 CLIENT TIER 24
3.3 MIDDLE TIER 25
3.3.1 Web Application and Logging database 25
3.3.2 Central Point Service (CPS) 25
3.3.3 Services Registry and Repository (SRR) 26
3.4 SERVICE PROVIDER TIER 26
3.5 LOCATION 26

4. BUSINESS PROCESS ARCHITECTURE 27

4.1 SEQUENCE OF PUBLISH, REQUEST, RESPONSE PATTERN 27

5. HARDWARE PLATFORM ARCHITECTURE 30

5.1 VIRTUAL PRIVATE SERVER (VPS) 30
5.2 PHYSICAL HARDWARE 30
5.2.1 Server capabilities 30
5.2.2 Storage capabilities 31
5.2.3 Service Provider capabilities 31

6. NETWORK ARCHITECTURE 33

6.1 HIGH AVAILABILITY 33
6.2 DATA CONFIDENTIALITY AND DATA INTEGRITY ..... **ERROR! BOOKMARK NOT DEFINED.**
6.3 MIDDLE TIER LOGICAL AND PHYSICAL CONNECTION 34

6.4 CLIENT AND SERVICE PROVIDER TIER LOGICAL AND PHYSICAL CONNECTION 35

7. SECURITY ARCHITECTURE 35

7.1 SECURITY SCOPE 36
7.2 SECURING WEB SERVICES 36
7.3 IDENTIFICATION AND AUTHENTICATION 36
7.4 AUTHORIZATION 36
7.5 ACCESS CONTROL 36
7.6 ACCESS AUDITING 36

8. END-TO-END SCENARIO 38

8.1 SCENARIO DESCRIPTION 39

9. TECHNICAL SPECIFICATION DOCUMENT 40

1. Introduction

Background: On 30th October 2009, a memorandum of understanding (MoU) has been signed between:

1. Ministry of Security represented by the Minister
2. B&H Border Police represented by the Director
3. State Agency for Investigation and Protection represented by the Director
4. Federation Police Administration represented by the Director
5. RS Ministry of the Interior represented by the Minister
6. B&H District Brcko Police Administration represented by the PA Head
7. Cantonal Ministries of the Interior represented by respective ministers
8. B&H High Judicial and Prosecutorial Council represented by the President

with the goal of establishing a system for electronic data exchange between police agencies and prosecutor offices. In the MoU it has been defined which databases will be available for exchange and which technology standard should be used.

The purpose of this system is to facilitate more efficient work of law enforcement bodies and prosecutor's offices and a better co-operation between law enforcement bodies and prosecutor's offices as one of the objectives set forth in the Road Map for accession to the EU and conditions for facilitation of a visa regime with the member countries of the Shengen Agreement.

1.1 Acronyms

AAA	Authentication, Authorization and Accounting
ACL	Access Control List
API	Application Programming Interfaces
AZLP	Agency for Personal Data Protection
B&H	Bosnia and Herzegovina
CPS	Central Point Service
EC	European Commission
ESB	Enterprise Service Bus
FB&H	Federation Bosnia and Herzegovina
FTP	File Transfer Protocol
GUI	Graphical User Interface
HDD	Hard Disk Drive
HTTP	HyperText Transfer Protocol

HTTPS	Hypertext Transfer Protocol Secure
HVM	Hardware virtual machine
IDDEEA	Agency for Identification Documents, Registers and Data Exchange of BiH
IIOP	Internet Inter-Orb Protocol
IPSec	Internet Protocol Security
LAN	Local area network
MEP	Message Exchange Patterns
MoS	Ministry of Security
MoU	Memorandum of Understanding
MSSQL	Microsoft SQL Server
MTOM	Message Transmission Optimization Mechanism
NAS	Network Attached Storage
NAT	Network Address Translation
QOS	Quality of service(QoS)
RMI	Remote Method Invocation
RMI-IIOP	Remote Method Invocation - Internet Inter-Orb Protocol
RPC	Remote Procedure Call
RS	Republika Srpska
SIPA	State Investigation and Protection Agency
SMTP	Simple Mail Transfer Protocol
SOA	Service Oriented Architecture
SOAP	Simple Object Access Protocol
SOAP	Simple Object Access Protocol
SQL	Structured Query Language
SRR	Services Registry and Repository
SSL/TLS	Secure Sockets Layer / Transport Layer Security
SW/HW	SOFTWARE/ HARDWARE
SwA	SOAP with Attachments’
UDDI	Universal Description, Discovery and Integration (UDDI, pronounced Yu-di
URL	Uniform Resource Locator
VM	Virtual Machine
VPN	Virtual Private Network
VPS	Virtual Private Server
WAN	Wide area Network
WS	Web Service

WSDL	Web Services Description Language
WSIL	Web Services Inspection Language
XML	Extensible Markup Language
XSLT	Extensible Stylesheet Language Transformations

1.2 Objective

The objective of this document is to provide the detailed design of a system which fully meets the conditions prescribed under the Agreement. The Agreement envisages the adoption of by-laws on contents and methods of record keeping which are the subjects of the Agreement. Hence, this document takes into account the adoption of pertinent by-laws as well.

The design of the system is an engineering document which defines technical functionality and methods of implementation of electronic data exchange related to law enforcement bodies and prosecutor's offices.

This document provides development guidelines for system deployment. A "proof of concept" is expected 1 month after the contract is signed. The whole project must be completed within 6 months.

1.3 Exclusions

It is important to note that High Availability will not be implemented end to end but in the Central Service Point only. To implement high availability in all access points, at least one more access link to the Synchronous Digital Hierarchy/SDH network has to be added and it needs to be ensured that network and hardware infrastructure in all agencies is redundant, which is not feasible in this moment.

1.4 Intended Users

The document is meant to be used by the contractor who will implement the system.

1.5 Architectural Goals and Constraints

The overall architecture goal of the system is to provide a highly available and scalable web-services exchange service for "need to know" users employed by agencies defined in chapter 1 first paragraph. Web-services exchange service has to enable users to understand what services are available for use.

The Electronic Data Exchange will be used:

- to exchange information between Agency requestor and Agency data provider defined by the MoU;
- to generate reports based on log data for data provider Agencies.

A key architectural goal is to leverage industry best practices for designing and developing a scalable, enterprise-wide web service exchange service. To meet this goal, the design of the Electronic Data Exchange will be based on core web service patterns as well as the industry standard development guidelines for building the Web service messaging.

1.6 Guiding Principles

Guiding principles provide a foundation upon which to develop the target architecture for the screening tool, in part by setting the standards and measures that the tool must satisfy. These in turn drive design principles that can be used to validate the design and ensure that it is aligned with worldwide Architecture, Design Principles and Standards.

Some of the guiding principles that will be followed during the Electronic Data Exchange and development are outlined below.

1.6.1 Scalable

Scalability is the ability of the platform to scale both up and down to support varying numbers of users or transaction volumes. The application should be able to scale horizontally (by adding more servers) or vertically (by increasing hardware capacity or software efficiency).

1.6.2 Flexible

Flexibility is the ability of the application to adapt and evolve to accommodate new requirements without affecting the existing operations. This relies on a modular architecture, which isolates the complexity of integration, presentation, and business logic from each other in order to allow for the easy integration of new technologies and processes within the application.

1.6.3 Standards-Based

Enterprise Service Bus services will comply with established industry standards. The standards-compliance will not only apply to service functionality but also to design, platform/infrastructure and other parts of the Electronic Data Exchange system. Examples of standards include HTTPS, XML, SOAP and WSDL.

1.6.4 Reliable

This quality of service measure ensures that business critical messages are not lost in transit. If the message is not delivered due to technical issues, the service request will be repeated. Platforms that support reliable message delivery include a series of acknowledgements, and a message resend capability guaranteeing that the message will eventually reach its destination. This WS-Reliable Messaging specification driven by Microsoft, Oracle, IBM, TIBCO, BEA and others defines a standard way of sending messages in a predefined sequence, and receiving an acknowledgement for receipt of those messages.

1.7 Web Services Definition

Web services are typically application programming interfaces (API) that can be accessed over a network, such as the Internet, and executed on a remote system hosting the requested services. Web services use Extensible Markup Language (XML) messages that follow the Simple Object Access Protocol (SOAP) standard. In such information systems, we have a machine-readable description of the operations offered by the service written in the Web Services Description Language (WSDL).

The W3C defines a "web service" as "a software system designed to support interoperable machine-to-machine interaction over a network". It has an interface described in a machine-processable format (specifically Web Services Description Language WSDL). Other systems interact with the web service in a manner prescribed by its description using SOAP messages, typically conveyed using HTTP with an XML serialization in conjunction with other web-related standards.

2. SOA Architecture and Topology

Service-oriented architecture (SOA) is a flexible set of design principles used during the phases of systems development and integration of the information system. SOA-based architecture provides a loosely-integrated suite of services in our case web-services that can be used within multiple police agencies.

SOA defines how to integrate widely disparate applications such as applications in different police agencies in FB&H and RS and prosecutor offices in Bosnia and Herzegovina which use multiple implementation platforms. SOA defines the interface in terms of protocols and functionality. An endpoint is the entry point for such an SOA implementation. Service-orientation requires loose coupling of services with operating systems, and other technologies that underlie applications. SOA separates functions into distinct units, or services, which developers make accessible over a network in order to allow users to combine and reuse them in the production of applications. These services and their corresponding consumers communicate with each other by passing data in a well-defined, shared format, or by coordinating an activity between two or more services.

The SOA architecture places into relationship various components and technologies that comprise a Web services "stack" or completely functional implementation. Valid implementations include subsets or parts of the stack, but must at least provide the components within the web service architecture. Because web service is one of the requirements of this project this document provides detailed relations between SOA architecture and web service functionality.

The SOA architecture includes Web services technologies capable of:

- Exchanging messages between clients
- Describing Web services Publishing and

- Discovering Web service descriptions

The Web services architecture defines an interaction between software agents as an exchange of messages between service requesters and service providers. Requesters are software agents that request the execution of a service. Providers are software agents that provide a service. Agents can be both service requesters and providers (peer to peer). Providers are responsible for publishing a description of the service(s) they provide. Requesters must be able to find the description(s) of the services.

The Web service architecture models the interactions between three roles: the service provider, the service discovery agency, and the service requestor.

The interactions involve publish, find, and bind operations. These roles and operations act upon the web service artifacts: the web service software module and its description. In a typical scenario a service provider hosts a network accessible software module (an implementation of a web service). The service provider defines a service description for the web service and publishes it to a requestor or service discovery agency. The service requestor uses a find operation to retrieve the service description from the discovery agency (i.e. a registry or repository or UDDI) and uses the service description to bind with the service provider and invoke or interact with the web service implementation. Service provider and service requestor roles are logical constructs and a service may exhibit characteristics of both.

Requesters and providers interact using one or more message exchange patterns (MEPs) that define the sequence of one or more messages exchanged between them. A service description is hosted by a discovery service, to which a provider publishes the description, and from which the requester discovers the description. The description includes data type and structure information, identifies the MEP, and contains the address of the service provider.

Software agents in the basic architecture can take on one or all of the following roles:

- Service requester -- requests the execution of a Web service (Web browser or application in Agency)
- Service provider -- processes a Web service request (Agency data owner)
- Discovery agency -- agency through which a Web service description is published and made discoverable (Directorate for coordination of police agencies)

A software agent in the Web services architecture acts in multiple roles, requester and provider and discovery agency. A service is invoked after the description is found, since the service description is required to establish a binding.

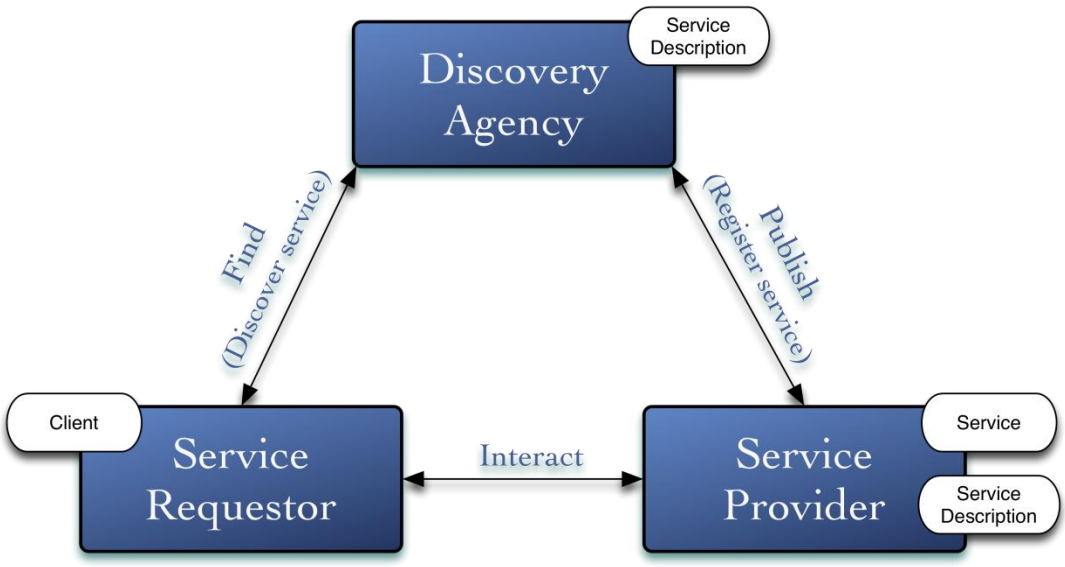


Figure 1

The figure above illustrates the relationships between requesters (Agency client), providers (Agency data owner), services (web service), descriptions (UDDI), and discovery services (– Directorate for coordination of police bodies) in the case where agents take on both requester and provider roles. For example, XML messages compliant with the SOAP specification are exchanged between the requester and provider. The provider publishes a WSDL file that contains a description of the message and endpoint information to allow the requester to generate the SOAP message and send it to the correct destination.

To support the common MEP of request/response, Web services implementation provides software agents that function as both requesters and providers, as shown in Figure 1. The service requester sends a message in the form of a request for information, or to perform an operation, and receives a message from the service provider that contains the result of the request or operation. The service provider receives the request, processes the message and sends a response. The technologies used for this type of Web services interaction include SOAP, WSDL, and HTTPS.

2.1 Components

The Service: Whereas a web service is an interface described by a service description, its implementation is the service. A service is a software module deployed on network accessible platforms provided by the service provider. It exists to be invoked by or to interact with a service requestor. It may also function as a requestor, using other web services in its implementation. This is our access point located in Agency data owner as shown in Figure 1.

The Service Description: The service description contains the details of the interface and implementation of the service. This includes its data types, operations, binding information, and network location. It could also include categorization and other metadata to facilitate

discovery and utilization by requestors. The complete description may be realized as a set of XML description documents. The service description is published to a discovery agency by the Agency where the service is located.

Roles

Service Provider: This is the owner of the service. From an architectural perspective, this is the platform that hosts access to the service. It has also been referred to as a service execution environment or a service container. Its role in the client-server message exchange patterns is that of a server.

Service Requestor: This is the Agency that requires a certain function to be satisfied. From an architectural perspective, this is the application that is looking for and invoking or initiating an interaction with a service. The requestor role can be played on several ways: by a browser driven, a program with or without a user interface, e.g. another web service or application. Its role in the client- server message exchange patterns is that of a client.

Discovery Agency: This is a searchable set of service descriptions where service providers publish their service descriptions. The service discovery agency will be centralized in the Directorate for coordination of police bodies. A discovery agency can support both the pattern where it has descriptions sent to it and where the discovery agency actively inspects the service provider for descriptions. Service requestors may find services and obtain binding information (in the service descriptions) during the development for static binding, or during the execution for dynamic binding. For statically bound service requestors, the service discovery agent is in fact an optional role in the architecture, as a service provider can send the description directly to service requestors. (More about Web Service Invocation see below).

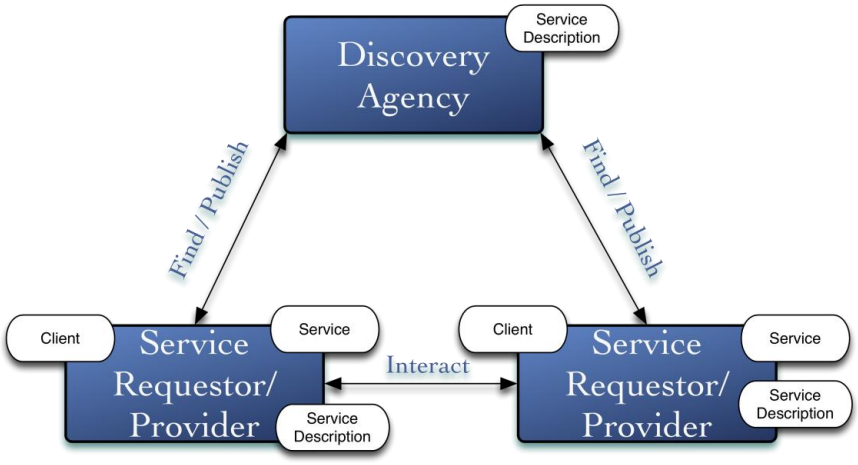


Figure 2

Figure 2 shows that Web service instances serve multiple roles simultaneously. In the peer-to-peer scenario, each peer Web service instance serves in both the Service Requester and Service Provider roles.

2.2 Operations

In order for an application to take advantage of Web services, three behaviors must take place: publication of service descriptions, finding and retrieval of service descriptions, and binding or invoking of services based on the service description. These behaviors can occur singly or iteratively, with any cardinality between the roles. In detail these operations are:

- **Publish:** In order to be accessible, a service needs to publish its description such that the requestor can subsequently find it. Where it is published can vary depending upon the requirements of the application (see Service Publication Stock discussion for more details).
- **Find:** In the "find" operation, the service requestor retrieves a service description directly or queries the registry for the type of service required (see Service Discovery for more details). The find operation may be involved in two different lifecycle phases for the service requestor: at design time in order to retrieve the service's interface description for program development, and at runtime in order to retrieve the service's binding and location description for invocation.
- **Interact:** Eventually, a service needs to be invoked. In the "interact" operation the service requestor invokes or initiates an interaction with the service at runtime using the binding details in the service description to locate, contact, and invoke the service. Examples of the interaction include: single message one way, broadcast from requester to many services, a multi message conversation, or a business process. Any of these types of interactions can be synchronous or asynchronous.

2.3 Web Services Stacks

To ensure interoperability when performing publish, find and bind operations expressed in the Service Oriented Architecture (SOA) diagram; conceptual and technical standards must be defined for each role and type of interaction. This section will explore each of the roles and interactions in order to identify each relevant stack of technologies.

2.3.1 Wire "Stack"

The wire stack encapsulates the concepts and technologies dealing with the actual physical exchange of information between any of the roles in the SOA diagram. This includes the variety of network transport, message packaging and message extensions that may be utilized to facilitate data exchange.

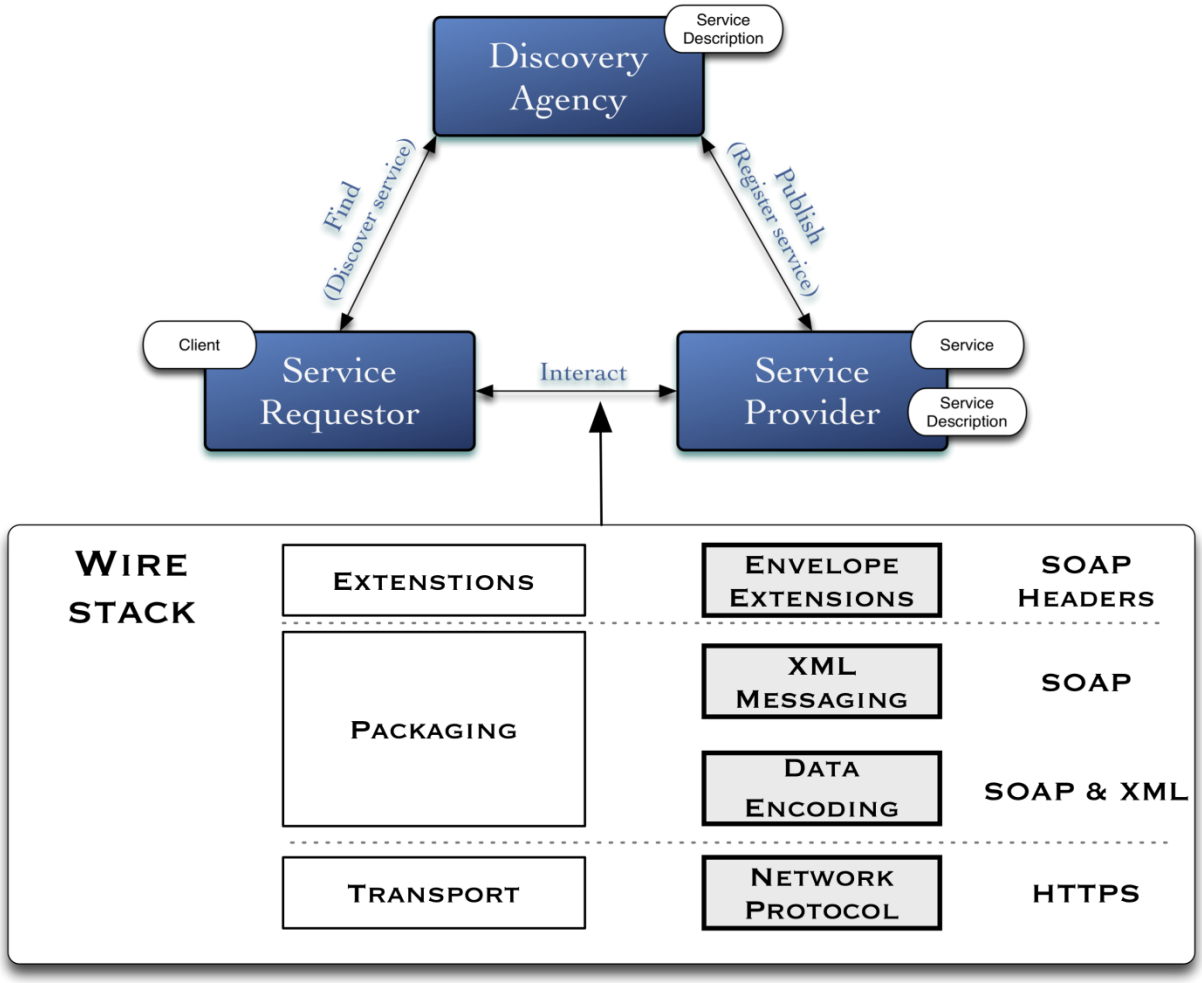

Service Description
Discovery Agency
Find (Discover service)
Publish (Register service)
Client
Service Requestor
Interact
Service Provider
Service
Service Description
WIRE STACK
EXTENSTIONS
ENVELOPE EXTENSIONS
SOAP HEADERS
PACKAGING
XML MESSAGING
SOAP
DATA ENCODING
SOAP & XML
TRANSPORT
NETWORK PROTOCOL
HTTPS

Figure 3

Transport

The foundation of the web services stack is the network. Web services must be network accessible to be invoked by a service requestor.

Among law enforcement agencies with multiple types of network infrastructures, HTTP can be used as a common, interoperable bridge to connect disparate systems. One of the benefits of web services is that it provides a unified programming model for the development and usage of private Intranet. As a result, the choice of network technology can be made entirely transparent to the developer and consumer of the service, this is further described in the Security Architecture chapter of this document.

Packaging

Packaging represents the technologies that may be used to package information being exchanged. XML has been broadly adopted as the basis for Web service message packaging protocols and is adopted for this project.

SOAP is a simple and lightweight XML-based mechanism for creating structured data packages that can exchange between network applications (web based or client/server applications). SOAP consists of four fundamental components: an envelope that defines a framework for describing message structure, a set of encoding rules for expressing instances of application-defined data types, a convention for representing remote procedure calls (RPC) and responses, and a set of rules for using SOAP with HTTP. SOAP can be used in combination with a variety of network protocols; such as HTTP, SMTP, FTP, RMI/IIOP, or a proprietary messaging protocol but for this project HTTP protocol has been chosen because of web service development simplicity.

The system should support the following protocols

- File / database adapter
- SOAP 1.1 / SOAP 1.2
- WSDL 1.1 / WSDL 2.0
- Mail transport (recommended for alert sending via email)
- WS-reliable messaging (allows messages to be transferred reliably between nodes)
- MTOM / SwA (suggestion for future exchange of binary attachments)
- WS-Security

Extensions

Building on the transport and packaging layers, the final layer in the Wire stack provides a framework that allows additional information to be attached to Web service messages representing a variety of additional concerns; such as context, routing, policy, etc. As a key part of its envelope message structure, SOAP defines a mechanism to incorporate orthogonal extensions (also known as features) to the message in the form of headers and encoding rules. It is expected that as Web services are adopted and evolved, a broad collection of such extensions will emerge and be standardized. Necessity of message envelope encryption putting this layer is the foreground.

2.3.2 Discovery Agencies "Stack"

While the bottom three layers of the stack identify technologies for compliance and interoperability, the service publication and service discovery can be implemented with a range of solutions.

Any action that makes a WSDL document available to a requestor, at any stage of the service requestor's lifecycle, qualifies as service publication.

Since a web service cannot be discovered if it has not been published, service discovery depends upon service publication. The variety of discovery mechanisms parallels the set of publication mechanisms. Any mechanism that allows the service requestor to gain access to the service description and make it available to the application at runtime qualifies as service discovery. This is usually the WSDL document obtained through a direct publish or the results of a previous find operation. Alternatively, the service may be discovered at design time or run time using a local WSDL registry, or a public or private registry such as UDDI. The variety of service discovery mechanisms is discussed in more detail in the section titled Service Discovery.

2.3.2.1 Service Publication - Producing Service Descriptions

The service description may be generated, hand coded, or pieced together based on existing service interface definitions etc. Developers may hand code the entire service description, including the UDDI entry. Tools exist to generate parts of the WSDL and potentially parts of the UDDI entry from metadata artifacts from the programming model and the deployment of the web service executable. Parts of the service description may already exist (for example the web service may be based on an industry standard service interface definition) such that very little needs to be further generated.

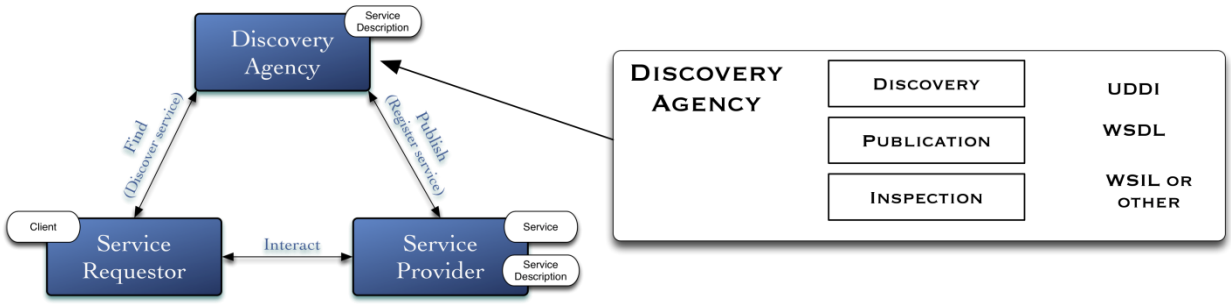


Figure 4

Publishing Service Descriptions

A service description (WSDL) will be published to Discovery Agency directly.

Publishing service descriptions available to Web services is through a UDDI registry.

Web services for use within law enforcement Agency for all applications clients should be published to a UDDI registry. The scope of this UDDI registry will be corporate and cover MoS and all Agency signatories of the Agreement. This UDDI registry sits behind the firewall and allows the service publishers more control over their service registry and its accessibility, availability, and publication requirements.

2.3.2.2 Service Discovery - Acquiring Service Descriptions

As with publishing Web service descriptions, acquiring Web service descriptions will vary depending on how the service description is published and how dynamic the Web service application is meant to be. Service requestors will find Web services during two different phases of an application lifecycle – design time and run time. At design time, service requestors will search for web service descriptions by the type of interface they support. At run time service requestors will search for a web service based on how they communicate or qualities of service advertised.

With the direct publish approach the service requestor will cache the service description at design time for use at runtime. The service description may be statically represented in the

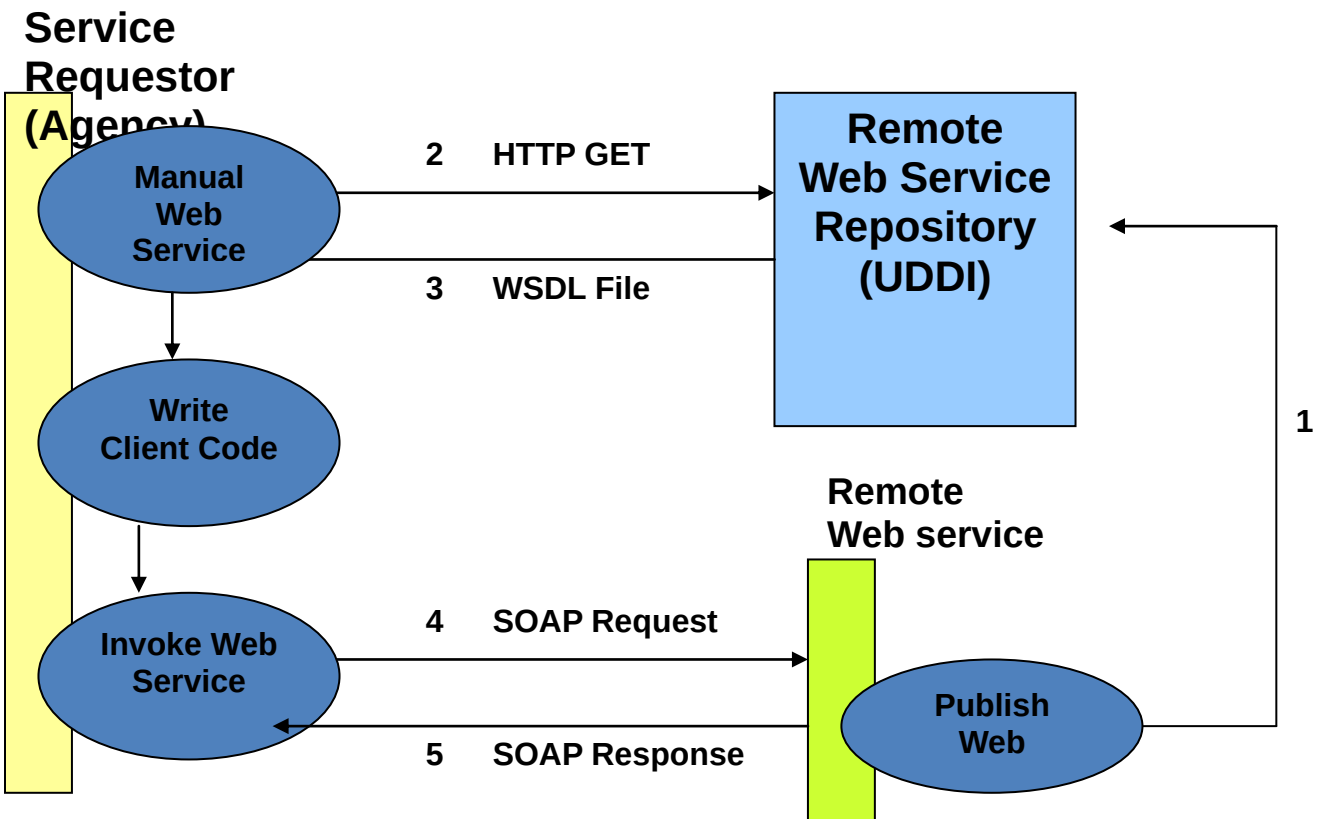


Figure 5

WSDL - Web Service Description
SOAP - Web Service Message Protocol

program logic, stored in a file, or in a simple, local service description repository see Figure 5.

Service requestors can retrieve a service description at design time or runtime from a Web page (URL), a service description repository, a simple service registry or a UDDI registry. The look-up mechanism will need to support a query mechanism that provides find by type of interface (based on a WSDL template), the binding information (i.e. protocols), properties (such as QOS parameters), the types of intermediaries required, the taxonomy of the service, business information, etc.

The various types of UDDI registries have implications on the number of runtime binding web services can choose from, policy for choosing one among many, or the amount of pre screening that must be done by the requestor before invoking the service. Service selection can be based on binding support, historical performance, and quality of service classification, proximity, or load balancing.

2.3.2.3 Consuming Service Descriptions

Once a service description is acquired, the service requestor will need to process it in order to invoke the service. The service requestor uses the service description to generate SOAP requests or programming language specific proxies to the web service. This generation can be done at design time or at runtime to format an invocation to the web service. Various tools can be used at design time or runtime to generate programming language bindings from WSDL documents. One of the most popular is Java programming language with various APIs and extensions. These bindings present an API to the application program and encapsulate the details of the XML messaging from the application.

Web Services Description Language: Describing Service Endpoints (Client - Service)

If no standards existed, it would have been difficult for web services and in turn SOA to be so widely accepted. WSDL provides the standardized format for specifying interfaces and allows for integration.

2.3.2.4 WSDL

WSDL forms the basis of web services and is the format that describes web services. WSDL describes the public interface of a web service including meta-data such as protocol bindings, message formats, and so on. A client wanting to connect to a web service can read the WSDL to determine what contracts are available on the web service. One of the key tenets of service orientation is that the services share contracts and schemas, not classes. As a result, when creating a service, it needs to be ensured that the contract for that service is well considered and is something that would not be changed.

A WSDL document has three parts, namely, definitions, operations, and service bindings, which can be mapped to one of the elements listed in Table 1.

Table 1. WSDL Document Structure

Element	Description
<portType>	Operations performed by the web service
<message>	Message used by the web service
<types>	Data types used by the web service
<binding>	Defines a communication endpoint (by means of protocol and address) to access the service
<Service>	Aggregate multiple ports (in combination with binding and address into a service)

Definitions

Definitions are expressed in XML and include both data type and message definitions. These definitions are based upon an agreed-on XML vocabulary that in turn should be based on a set of industry-wide vocabulary. If it is needed to use data type and message definitions between organizations, then an industry-wide vocabulary is recommended. The project implementer has to choose one of the Schema design patterns as follows: Russian Doll, Salami Slice, or Venetian Blind.

Operations

Operations describe the actions for the message supported by the web service and can be one of four types, as listed in Table 2. In a WSDL document structure, operations are represented using the <portType> element, which is the most important element because it defines the operations that can be performed. In the context of the OO paradigm, each operation is a method.

Table 2. Operation Types

Error	Description
One-way	The service endpoint Agency provider receives a message.
Request-response	The service endpoint Agency provider receives a message and sends a correlated message to Agency requestor.
Notification	The service endpoint Agency provider sends a message.

2.3.3 Service Bindings

Service bindings connect port types to a port (that is, the message format and protocol details for each port). A port is defined by associating a network address with a port type. A service can contain multiple ports. This binding is commonly created using SOAP. The binding element has two attributes: a name that can be anything to define the binding and the type, which points to the port for the binding.

2.4 Encoding

Data will be exchanged between heterogeneous systems therefore a common representation must be envisaged. Example 1 illustrates that a single piece of data like a telephone number may be represented in many different and equally valid ways in XML.

Example 1. Many XML representations of a phone number

```
<phoneNumber>(123) 456-7890</phoneNumber>
  <phoneNumber>
<areaCode>123</areaCode>
<exchange>456</exchange>
<number>7890</number>
</phoneNumber>
<phoneNumber area="123" exchange="456" number="7890" />
<phone area="123">
<exchange>456</exchange>
<number>7890</number>
</phone>
```

The following needs to be defined:

- The types of information to be exchanged
- How that information is to be expressed as XML
- How to package and send the information

Standards

Any characters outside the range of characters that can be included in the document must be escaped and identified as character references. The escape sequence used throughout XML uses the ampersand (&) as its start and the semi-colon (;) as its end. The syntax for character references is an ampersand, followed by a pound/hash sign (#), followed by either a decimal character code or lowercase x followed by a hexadecimal character code, followed by the semicolon. Therefore, the 8- bit character code 128 will be encoded in a UTF-8 XML document as €.

Without these agreed conventions, programs cannot know how to decode the information they're given, even if it's encoded in XML. SOAP provides these conventions.

Specific project solution will offer support for various XML Encoding standards but only one standard should be adopted and implemented in this project.

Recommended features are:

- Message transformation
- Protocol transformation (standard with ESB)
- Load balancing (for clustering purposes)

- Fail-over sending (in case of failure at central bus, another bus is sending)
- Schema validation (for possible rejection of incorrect messages)
- Logging and monitoring
- Content based routing (and other mediation services according to EIP)
- Service Orchestration (BPEL at message level, BPMN at process level)
- Rules engine
- Graphical design interface for orchestration, mediation and configuration
- Key store (for administration of certificates)
- User administration
- ESB expandability (related to programming / procedure languages)

3. Logical design

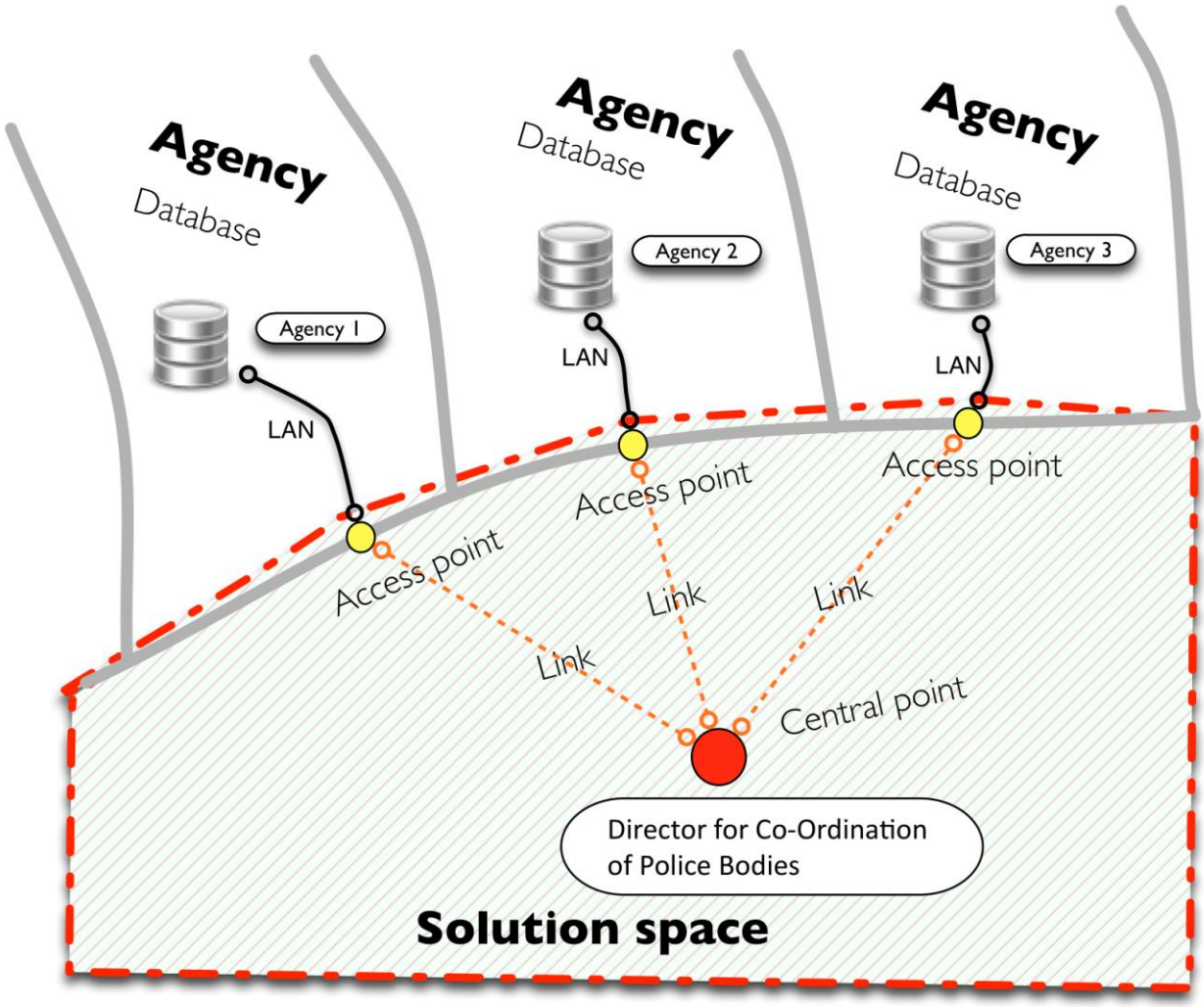


Figure 6

The diagram above provides an illustration of the Electronic Data Exchange project solution space along with various geographic locations that will be used in architecting of this solution.

Interaction of software components along with its responsibilities is explained below (Figure 7):

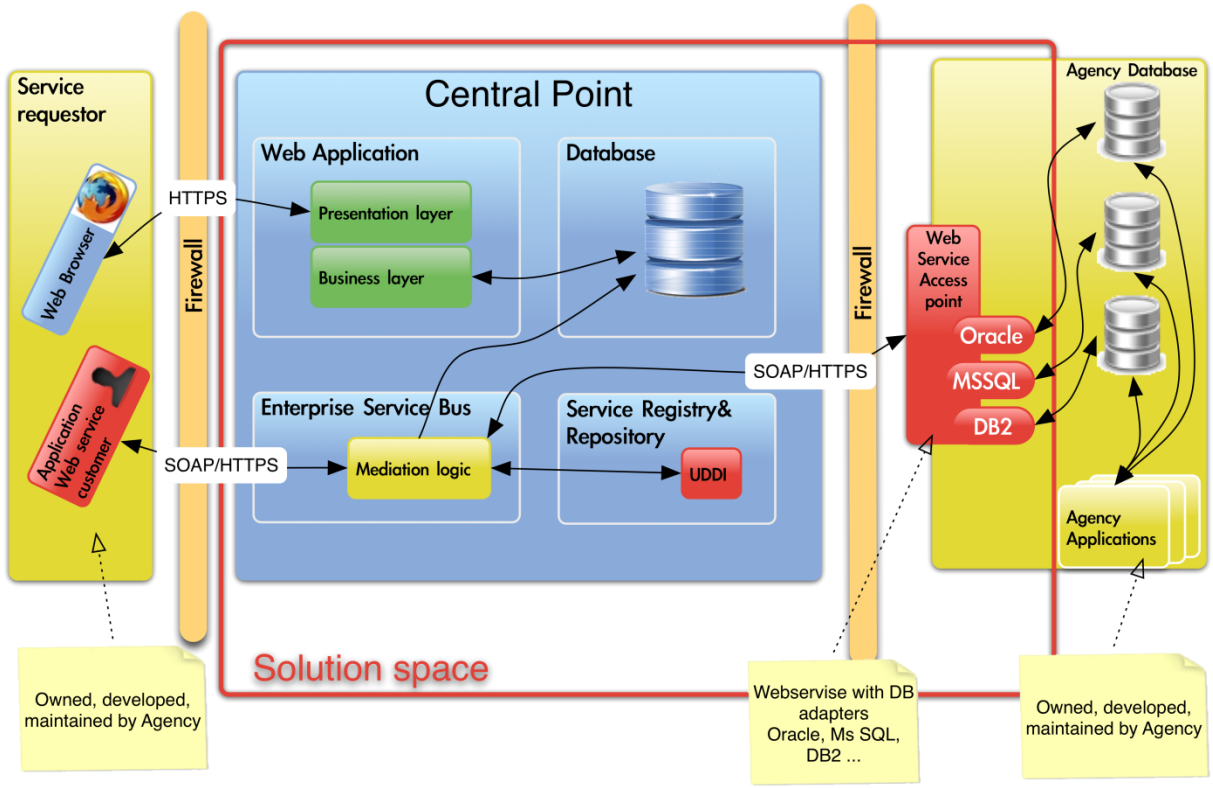


Figure 7: Architecture overview diagram

3.1 Capacity

3.1.1 Central Point Service (Enterprise Service Bus and WS Repository)

Capacity Min 100 SOAP WS
Min 100 Users
Number of Agency 7-20
Concurrent requests minimum 10
Service Registry & Repository with UDDI.
Central Point Service has to be implemented with unlimited numbers of virtual servers.
All virtual servers must be located on two physical servers.

3.1.2 Elementary Web Services

Number of access points will be between 7 and 20.

3.2 Client Tier

Two types of clients will have access to the “Central-point”. The first type of client is an **application client** (web service client), which will allow to request services, exposed by central

point using SOAP over HTTP. The second type of client is a **Web browser**. It will allow users to access GUI part of a solution (view log files, events, simple administration and configuration).

3.3 Middle Tier

"Central point" can be viewed as a broker between service requesters and service providers. There are three main sub-systems, which constitute the middle tier. And they are:

- Web Application and Logging database
- Central Point Service
- Services Registry and Repository

3.3.1 Web Application and Logging database

Web application and logging database are used to log all requests without entering in to payload of the requests and response. Logging database should record information about requester and Web service that has been requested, date and time stamps should be recorded also. If there is response information about it should be recorded as well.

Initial retention time of log data has to be 36 months. Consultation of data protection authority (AZLP) is pending and decision will specify final data retention time.

3.3.2 Central Point Service (CPS)

CPS manages the flow of messages between service requesters and service providers. *Mediation modules* within the CPS handle mismatches between requesters and providers, including protocol or interaction-style, interface and quality of service mismatches.

Mediation components operate on messages exchanged between service endpoints. In contrast with regular business application components, they are concerned with the flow of the messages through the infrastructure and not just with the business content of the messages. Rather than performing business functions, they perform routing, transformation, and logging operations on the messages. The information that governs their behavior is often held in headers flowing with the business messages.

External clients send web service requests (SOAP/HTTP requests) to **CPS**. Mediation logic is responsible to log the service request to database.

Mediation logic also makes a dynamic endpoint selection – it determines which external web service(s) should be called. After mediation logic determines, which services should be called, it makes a specific number of SOAP requests to certain service providers. Response from each service provider is stored temporary and finally results are aggregated and returned back to the service requestor in a SOAP response message.

Administration of Central Point (ESB) will be able to manage rules which will determine which application client can access to which service provider. This information will be stored in the

database and managed using GUI part of the solution. Central Point Service maintains its state in its own database.

3.3.3 Services Registry and Repository (SRR)

Service Registry and Repository (SRR) is a system for storing, accessing and managing information, commonly referred as service metadata, used in the selection, invocation, management, governance and reuse of services in a successful service-oriented architecture (SOA). For example, it is where information about services in the systems is stored, or in systems from other organizations, that are already used, are planned to be used, or it is important to be aware of. For example, an application can check the SSR just before invoking a service to locate the service instance best satisfying its functionality and performance needs. SRR also plays a role in other stages of the SOA life cycle.

Each web services, exposed by any external entity (organization) in the system, will be published to SRR. SRR thus allows web services easier to manage, use and maintain.

3.4 Service Provider Tier

“Central-point” implicitly contains knowledge on which external web service should be called. External web services are provided by organizations, which have the responsibility to maintain certain data.

In an ideal case, external organization will expose its web services according to some predefined web service specification, described by WSDL. It will then register this service to the SRR, which is a part of a solution and is positioned at the central point.

It is assumed, that not all external organizations will expose their own web services, but will at least provide a database, which could be transformed into service provider and participate in a service oriented architecture.

3.5 Location

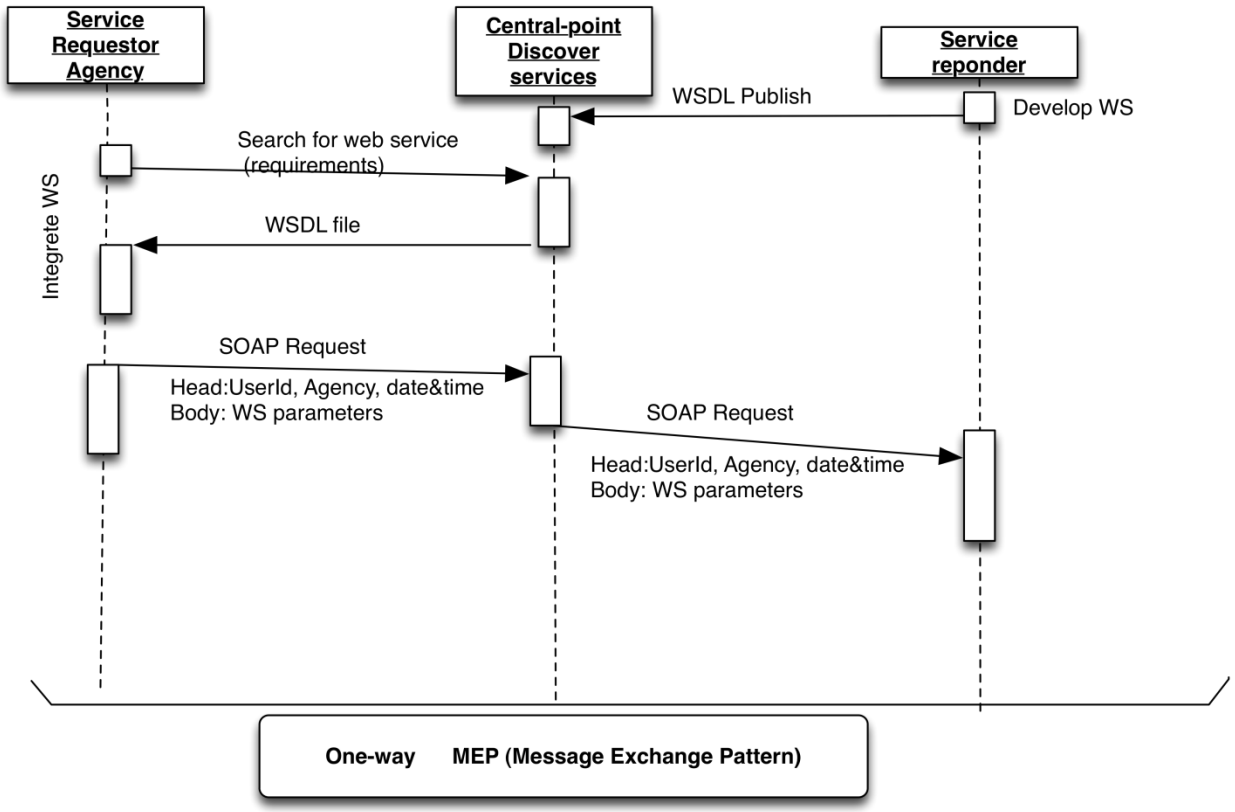
The Central Service Point will be located in a separate room in the National Operational Center/NOC2 (Trg Bosne i Hercegovine 1, Sarajevo) and will be under control and management of the Directorate for Co-ordination of Police Bodies. If the Directorate for Co-ordination of Police Bodies decides to physically move the Central Service Point, this has to be possible without long service interruption.
Equipment needed for the implementation of the Service Provider Tier will be located at the location decided/provided by agencies that are participating in the Data exchange.

4. Business process architecture

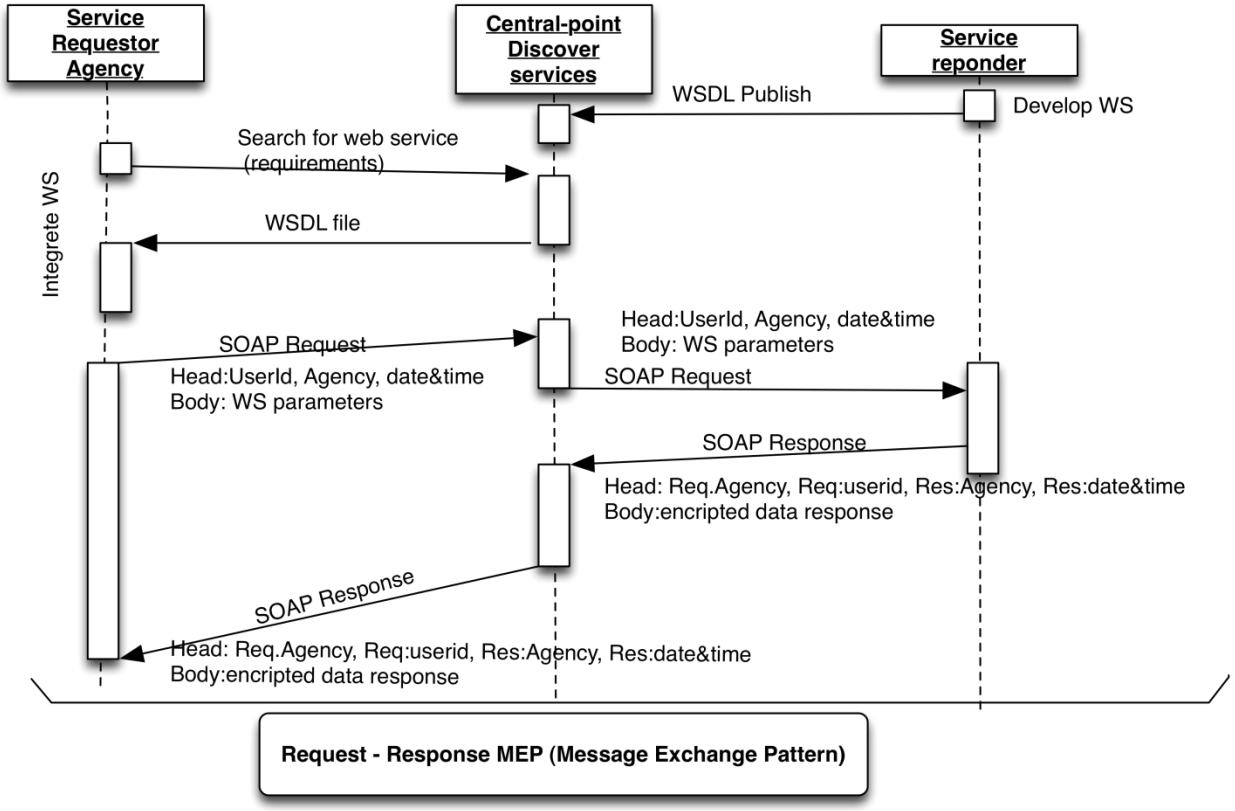
4.1 Sequence of publish, request, response pattern

In this section the significant interactions between the major components for the system modules from a web service description publishing to executing of web service using the central point discovery service engine are detailed.

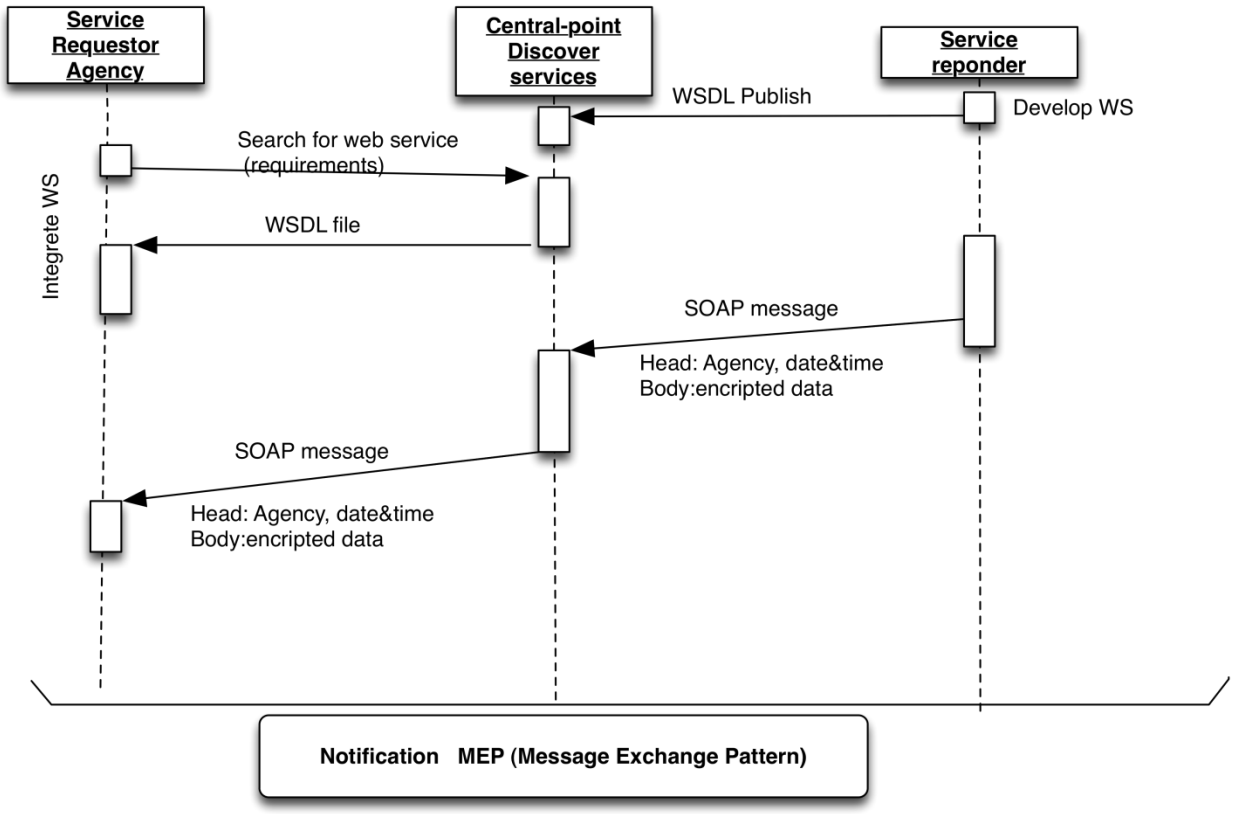
Note: The Sequence Diagram below is only a logical representation of the significant interactions of the system and may not directly map to the physical interactions of the system.



One-way MEP (Message Exchange Pattern)



Request - Response MEP (Message Exchange Pattern)


Service Requestor Agency
Central-point Discover services
Service reponder
WSDL Publish
Develop WS
Search for web service (requirements)
Integrete WS
WSDL file
SOAP message
Head: Agency, date&time
Body:encripted data
SOAP message
Head: Agency, date&time
Body:encripted data
Notification MEP (Message Exchange Pattern)

5. Hardware platform architecture

5.1 Virtual Private Server (VPS)

In order to achieve scalability and redundancy the hardware platform will be based on VPS (Virtual Private Server) technology. This also achieves the goal of an economical hardware solution.

VPS technology used for this system has to provide the following features:

1. Hypervisor has to be based on Intel-V and/or AMD-VT technologies
2. Consequently, it has to support hardware assisted Virtualization (HVM), para-virtulization is not an option for this system
3. Guest operating system, both Windows and Linux based, has to be able to run unmodified on VPS
4. Centralized management console for VPS administration
5. Live migration of virtual machines across physical hosts
6. Creation of VM backup images and restoring from those images
7. Support for NAS storage

Additional features supported by VPS technology used for this system are:

8. Support of virtual machines templates, i.e. possibility to create virtual machine based on predefined template (appliance)
9. Possibility to assign physical processor cores to a virtual CPU (i.e. 1 virtual CPU occupies resources of 1 core on physical CPU)
10. Storage management utilities or console

Virtualization system should provide transparent support for network storage to VMs (i.e. space for virtual HDD can be allocated on NAS).

5.2 Physical hardware

The supplier is obliged to provide system components (servers, network switches, data storages etc.) according to the given specifications. All hardware has to be powered with redundant power supplies built in.

5.2.1 Server capabilities

In order to achieve scalability, redundancy and price level for this system, there should be two physical servers with following minimal requirements described in chapter *10. Technical Specifications* under item 2.1.

These two (minimum) servers have to be connected through GB Ethernet connection between them in order to achieve fast failover. Servers will be configured in cluster with two virtual servers. One of the servers will serve as a primary and the other as a secondary or backup for failover. In case of any VPS crashing or becoming compromised it is required for the system to provide means (i.e. direct Ethernet link, remote management console etc.) of activating backup VPS on backup server.

Another variant is to distribute primary VPSs between the two physical servers thus distributing load. This also provides redundancy in case of failure of one of the physical servers, where backup VPSs take place of the primary ones that were running on the crashed physical server. The design team leaves it to the bidder to propose high availability solution on application layer.

5.2.2 Storage capabilities

Primary storage has to be SAN or NAS connected to the physical servers via iSCSI or Fiber Channel (iFCP).

Network storage appliance has to have the features described in detail in chapter *10. Technical Specifications* under item 2.2.

5.2.3 Service Provider capabilities

Integration and security solution plays a critical role in this scenario. It is hardware based solution - appliances preferred - which includes Accelerator (XML and XSLT Accelerator), Security Gateway (AAA, WebService Proxy, XML Firewall, Policy Enforcement (Security and Governance), Encryption / decryption, Signature Verification ....) and Integration Gateway (Data transformation and protocol switching) in one box. Integration and security solution will be placed at service provider tier and will:

- Be used as a security gateway to the external system
- Provide web service interface to the external system
- Provide AAA functionality (Authentication – Authorization and Accounting)
- Provide hardware based XML acceleration for SOAP message parsing
- Provide hardware based XML encryption and XML signature
- Provide support for PKI infrastructure
- Provide support for integration with Service Registry and Repository

Solution at every service provider will be connected to the database via LAN, and will allow us to expose database functionality as a web service. This allows service requestors to pass request parameters in a SOAP message, and then map them to input parameters of the SQL statement or SQL procedure. After obtaining a result set from a database, it is then mapped into SOAP response and sent back to the service requestor. Supported databases for this scenario must be:

- Oracle
- DB2
- Microsoft SQL
- Sybase

Integration and security solution must provide fast and flexible development of web services. The requested integration and security solution should be implemented, deployed and functional for the exchange of web service messages **within 3 months** from the start of the project.

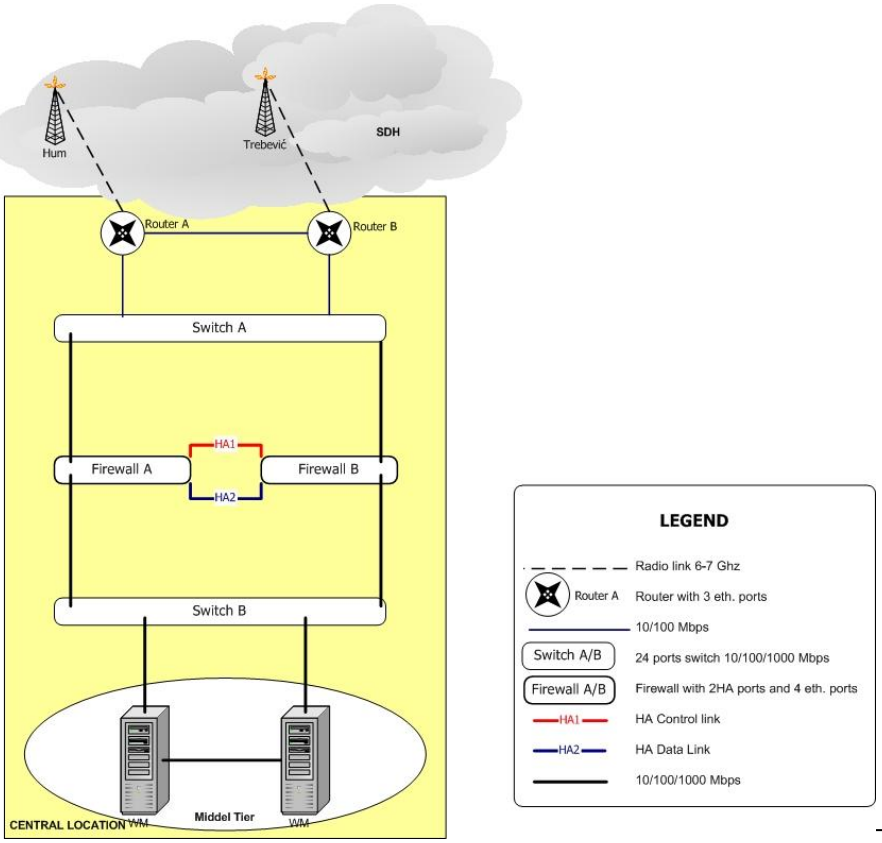
Some external systems may require user credentials for authentication and authorization. As mentioned earlier, username and a password will be contained in a SOAP header and will be extracted by an appliance if required by the external system.

6. Network Architecture

In order to establish reliable communication between agencies, central point and data service providers a reliable communication network must be used. Currently all agencies participating in the project are connected to the "B&H State" SDH network on layer 2 level. In order to save cost it has been decided that all data transmission will be performed via this network. For the purpose of this design document it is assumed that the "B&H State" SDH network is sufficiently reliable and with enough spare capacity to support data transmission or data exchange. Even though the payload of the XML messages will be encrypted it has been decided to establish IPSec VPN tunnel between middle tier on one side and client tier and service provider tier on the another side. IPSec gateway will have on WAN side IP addresses assigned by the Agency for Identification Documents, Registers and Data Exchange of BiH (IDDEEA). It is recommended to request /27 address block which will be used for this project only. IPSec gateway on LAN side will have LAN IP address. In order to ensure that there is no address overlapping in middle tier component of the system, and to enable unique identification of clients on clients side IPSec VPN gateway will have to implement one to one NAT. Address pool for one two one NAT will be provided by administrator of the middle tier component.

6.1 High availability

Based on anticipated volume of data that will be exchanged in the first year of system use it has been decided to implement high availability network infrastructure only in middle tier location.



Client/service provider locations will not satisfy high availability requirements. If during exploitation system administrators find that some of service provider locations are becoming more and more mission critical for clients, high availability architecture will be introduced on case by case bases.

In middle tier location two redundant communication links will have to be installed connecting middle-tier switches with the SDH

network in two different core locations.

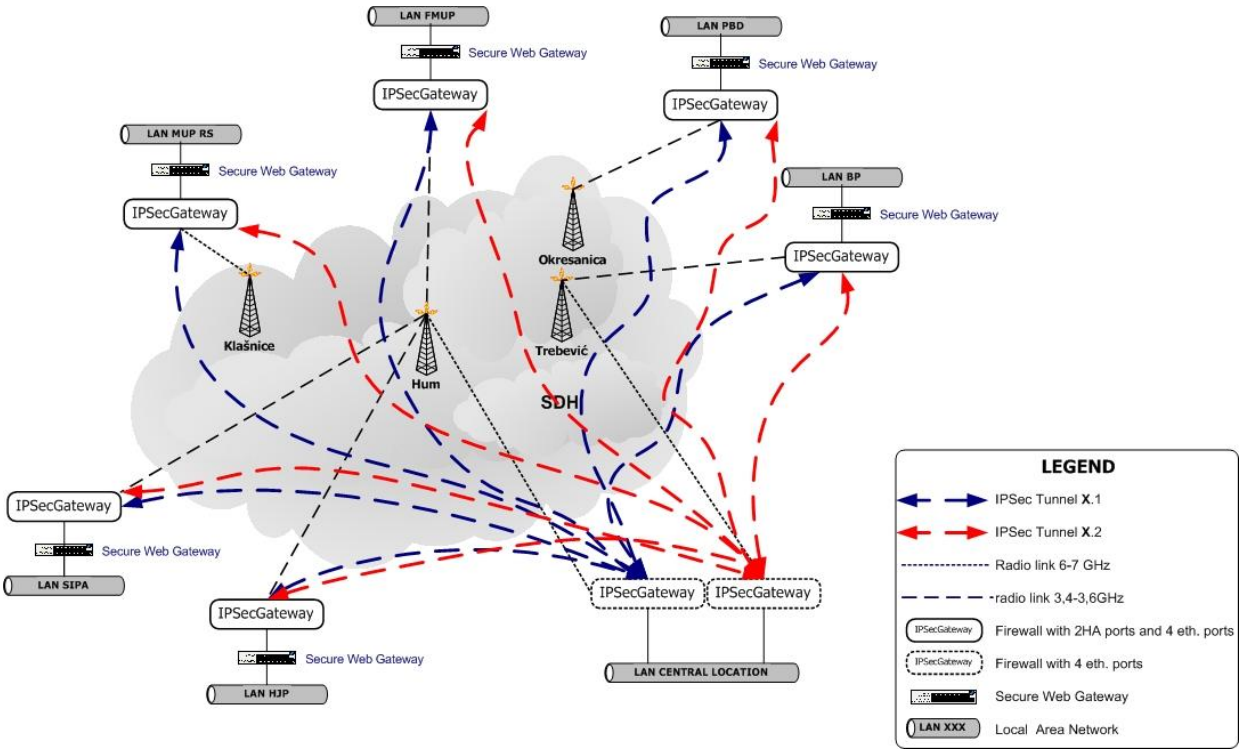
6.2 Data confidentiality and data integrity

Confidentiality and integrity of data transmitted via the "B&H State" SDH network will be ensured by using IPSec tunnels. It is important to note here that IPSec gateway devices have to be configured in "Fail safe" mode, in order to prevent leakage of information in case of device failure.

6.3 Middle tier logical and physical connection

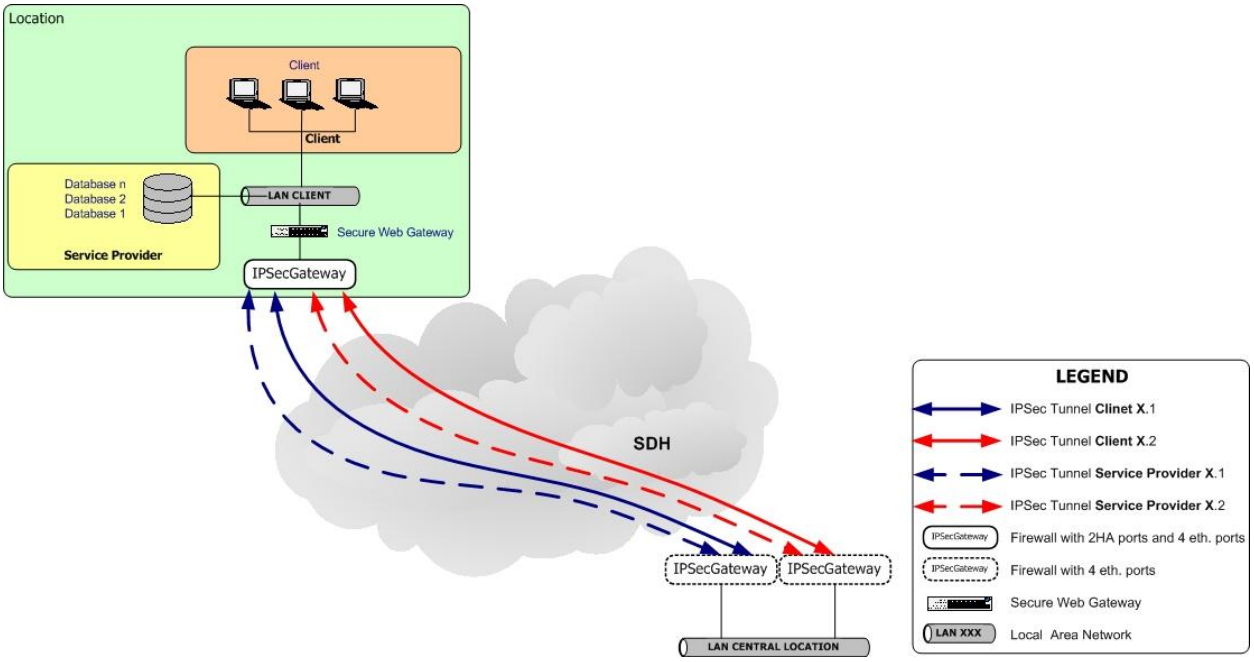
Middle tier equipment is connected to the network with two independent radio links @5-7GHz to SDH core network. Primary link is to Hum and secondary/backup link is to Trebevic (see picture below). Connection is established via Layer 3 routers. Router A and Router B are connected with 100 Mbps link. OSPF is used to insure failover. Router A and Router B are connected to 24 ports switch A. Switch A is also connected to Firewall A and Firewall B. Each Firewall has two HA (High Availability) ports, used to synchronize firewall states. Firewall trust side is connected to 24 ports switch B. Switch B is connected to redundant VM servers.

All communication with service providers and clients middle tier logic will have to tunnel through two IPSec gateways configured in failover mode and connected to two different communication links. Routing will have to be implemented in the IPSec gateways in order to detect link state and to insure uninterrupted failover.



6.4 Client and service provider tier logical and physical connection

In case where single LAN contains databases with data for exchange and clients work station network design will have to separate those two functions.



7. Security Architecture

From information security perspective this project has to achieve two very important security objectives. Firstly SW/HW solution needs to protect dialog between provider and client while satisfying requirement of recording provider and client info in middle tier databases. In other words security mechanisms implemented in web services and SOA have to satisfy well-known security dimensions:

- **Integrity**: message must remain unaltered during transmission across all intermediary services, such as network devices and software components.
- **Confidentiality**: contents of a message cannot be viewed while in transit, except by authorized services that need to see the message contents in order to perform routing.
- **Availability**: message is promptly delivered to the intended recipient, thus ensuring that legitimate users receive the services they are entitled to.
- **Non-Repudiation**: requestor should not be able to deny that he has requested service and provider should not be able to deny that service has been provided by him.

Secondly security mechanisms to protect infrastructure element at middle tier and provider side from unauthorized view or modification need to be designed. This will be done on device by device basis using well known techniques such as host hardening, enforcing strong password polices or applying ACL.

7.1 Security scope

Protecting provider and requester internal infrastructure such as work stations and databases is not in the scope of this document. It is assumed that above mentioned infrastructure is well protected and trusted, even though different service providers may have different security policies and management procedures.

7.2 Securing web services

Since SSL/TLS provides confidentiality at the transport layer only, XML Encryption provides confidentiality at the application layer and thus assures end-to-end confidentiality of messages traversing multiple Web services. In order to satisfy requirements on confidentiality of the dialog defined in the MoU, XML Encryption should be implemented at service provider side. XML encryption defines a standard model for encrypting both binary and textual data, as well as the means for communicating the information needed by recipients to decrypt the contents of received messages. Pre-shared secrets and IPSEC VPN tunnels will be used to insure that only authorized requestors can access to Central service.

7.3 Identification and authentication

The Central point is not responsible for individual user identification and authentication. If this will be required at the service provider side, than the appliance (on the service provider side) has to exchange security information with the provided user registry and authenticate the user using username and password from the message header.

7.4 Authorization

The Central point is not responsible for individual user authorization. If this will be required, the service provider side device has to exchange security information with the provided user registry and authorize the user based on credentials, stored in the user registry from the service provider.

7.5 Access Control

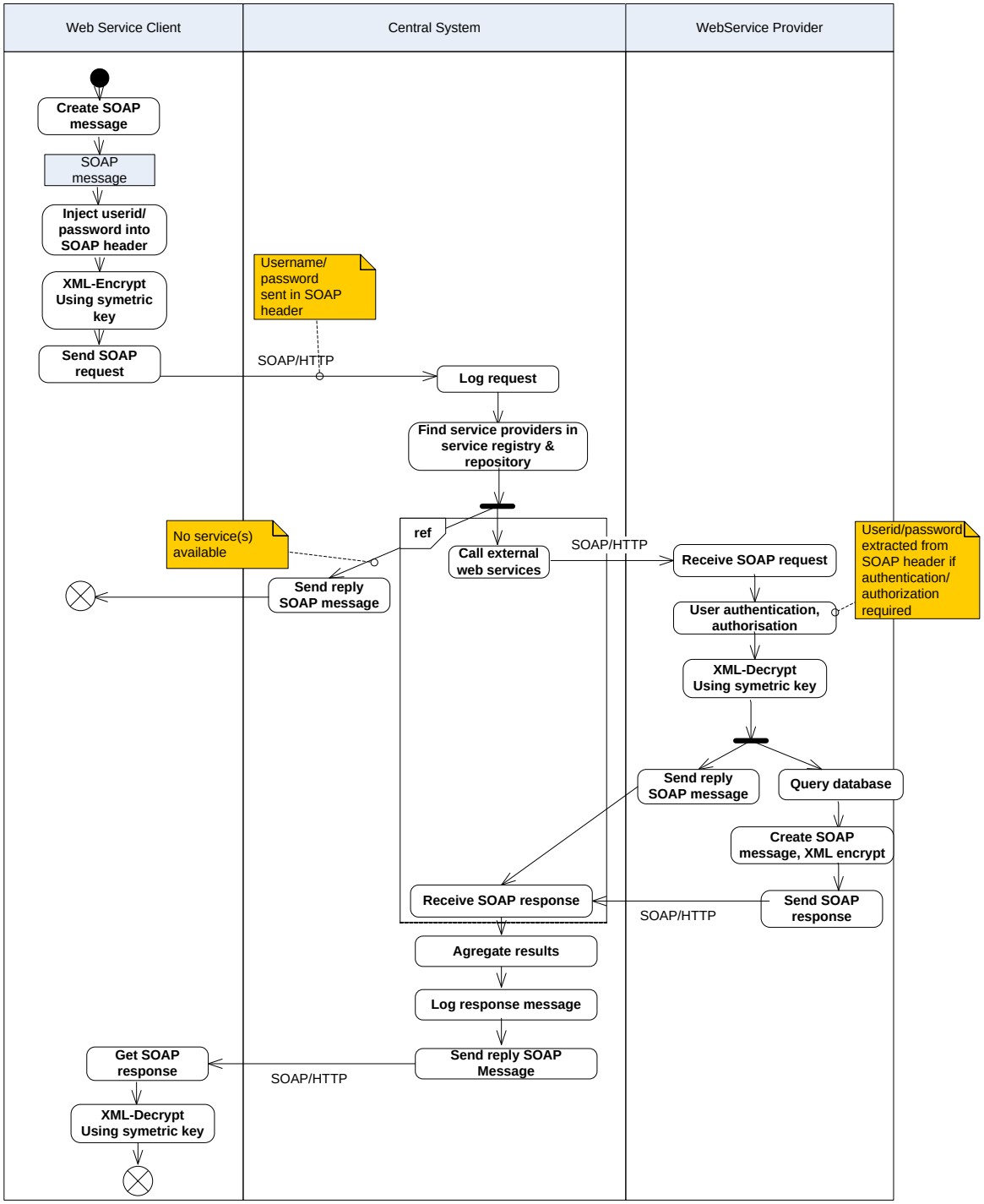
The Central point is not responsible to authorize access to web services on a per user bases. The access to the Central point will be restricted to trusted IP addresses with access to the Central point trough VPN and the state owned SDH network.

7.6 Access Auditing

Access to the Central Point Services will be logged in the Central Point database. Initial data retention time is36 months. Consultation of the Agency for Protection of Personal Data (AZLP) is pending and its decision will specify the final data retention time. Part of the solution is also a

Web Application, which will allow easy and quick monitoring of web service calls in the Central Point. Appliance (on Service provider side) should also provide separate logging capabilities.

8. End-to-end scenario



8.1 Scenario Description

Typical scenario starts, when a user of a certain client wants to use a service, provided by a »central-point«. A SOAP message is created by the service requestor application according to some predefined WSDL. User credentials (userid and password) are inserted into SOAP header in the same way as specified by WS-Security specifications.

The main benefit of this approach is that it allows an easy transition to PKI based security in the future with small code changes. In a PKI based solution, each user will be uniquely identified by a certificate, containing a private and a public key, that can be used for securing the transport channel as well as digitally signing a SOAP message. WS-Security specification allows passing user certificates in a SOAP header just the same way it allows passing username and password. Some web services may require that a certain part of a SOAP message is encrypted. Encryption is performed by a symmetric key, which should be kept private by each organization. SOAP encrypted message is then sent over HTTP protocol to the central point, which performs logging to the database. Any regular user should not access the database directly. Since the data, that the service requestor is looking for may reside in many external systems, the central point determines, which external services it should call to obtain this data. Information on external services is found in a database.

The central-point performs certain number of SOAP based calls to external web services to obtain the information required.

The Web service provider may or may not require the user to authenticate. User credentials are available in a SOAP message header. After reception of the SOAP message at the service provider side, the parts of a message are decrypted using symmetric key. SOAP message parameters are mapped to SQL statement parameters. After querying the database and obtaining the result set, the SOAP response message is created and all necessary fields are encrypted. The response is sent back to the central point.

If several external web services should be called, the central point aggregates the final result and returns the SOAP response back to the calling service requestor.

The Service requestor upon reception of the web service reply decrypts the XML message using symmetric key.

9. General requirements

SYSTEM INTEGRATION, CONFIGURATION, TESTING AND PROJECT MANAGEMENT

1. Proof of concept

It is expected from the contractor to deliver a proof of concept **within one month** after the contract is signed. This includes an example service requestor, an example service provider and an example central access service. All possible project obstacles have to be identified within this time frame. According to the tender documentation, problem identification after that point of time with impact on the project implementation time frame may lead to penalty fees.

2. Integration and security solution

It is expected from the contractor to demonstrate and prove the functionality of the requested integration and security solution to the service providers for the exchange of web service messages **within three months** from the start of the project.

3. Installation

It is expected from the contractor to deliver install and configure all elements needed for the Central Service point, as proposed in the build document.

It is expected from the contractor to deliver install and configure all elements needed for an access point, as proposed in the build document.

The beneficiaries in the location of the Central Service point and in the location of the access points will provide adequate administrative and building support.

The project installation must be completed **within six month** after the contract is signed.

4. Configuration

It is expected from the contractor to create at least six (6) web services for each agency and to demonstrate capabilities of functionality described in the detailed design document.

NOTE: Due to sensitivity of data fields included in web services exact web services will be given to the successful bidder. Project configuration shall be completed within six month after the contract is signed.

5. System Test

It is expected from the contractor to provide a system test document for all functionalities described in the detailed design document. Execution of the test cases will be done by an implementation team comprising representatives of the beneficiary institutions with the assistance of the contractor. The System test document will be reviewed by the implementation team and approved by the contracting authority.

6. Training

It is expected from the contractor to provide a minimum of 10 (ten) days of training on maintenance and support of the systems for minimum 4 (four) Administrators.

7. On-site support

The contractor is required to provide on-site support at least a year in production time.

8. Warranty

On-site warranty for all hardware and software is 1 (one) year.

Maximum response time to repair damaged equipment, delivered by manufacturer's service network is the next business day.

9. Project Management

It is expected from the contractor to implement a detailed Project Plan and provide manpower to execute it.

10. Security

The contractor has to ensure the security of the System from potential misuse during delivery, installation and handover to the Directorate for Coordination of police bodies.

10. *ANNEX II + III:* TECHNICAL SPECIFICATIONS + TECHNICAL OFFER

Publication reference: EuropeAid/129821/C/SUP/BA

Column 1-2 should be completed by the Contracting Authority
Column 3-4 should be completed by the tenderer
Column 5 is reserved for the evaluation committee

Annex III - the Contractor's technical offer

The tenderers are requested to complete the template on the next pages:

- Column 2 is completed by the Contracting Authority shows the required specifications (not to be modified by the tenderer),
- Column 3 is to be filled in by the tenderer and must detail what is offered (for example the words “compliant” or “yes” are not sufficient)
- Column 4 allows the tenderer to make comments on its proposed supply and to make eventual references to the documentation

The eventual documentation supplied should clearly indicate (highlight, mark) the models offered and the options included, if any, so that the evaluators can see the exact configuration. Offers that do not permit to identify precisely the models and the specifications may be rejected by the evaluation committee.

The offer must be clear enough to allow the evaluators to make an easy comparison between the requested specifications and the offered specifications.

Item No.	Quantity	Specifications Required	Specifications Offered	Notes, remarks, ref to documentation	Evaluation Committee's notes
1		**SOFTWARE/LICENSES**			
1.1.	**1**	**Central Point Service (Enterprise Service Bus and WS Repository)** Must provide standard Web Service, Protocols (HTTPS, SOAP, and XML). The proposed solution has to be in line with the design principles defined in the Detailed Design Document. The Central Point Service can be implemented with unrestricted numbers of virtual servers. All virtual servers must be located on minimum two physical servers. **Message Security** XML messages have to be protected from snooping using encryption and from unauthorized changes using authentication hashing. The proposed solution for the Central Point Service and message security has to be in line with the design principles defined in the Detailed Design Document.			
1.2.	**7**	**Elementary Web Services** Elementary Web Services have to establish on-line access to the data sources. This requirement applies in particular to the connection between the access points within the Agencies and their internal databases. Web Services can be implemented with server(s) or special device(s). Direct access to databases: Oracle, Microsoft SQL, Sybase and IBM DB2 is required. The proposed solution has to be in line with the design principles defined in the Detailed Design Document. **Message Security** XML messages have to be protected from snooping using encryption and from unauthorized changes using authentication hashing. The proposed solution for Elementary Web Service and message security has to be in line with the design principles defined in the Detailed Design Document.			
1.3	**1**	**Server Virtualization** Server Virtualization product comparable to: VSphere 4.1 ESXi enterprise plus or equivalent vCenter Site Recovery Manager 4 or equivalent Necessary licenses according to the servers offered under 2.1			
2.		**HARDWARE**			

Item No.	Quantity	Specifications Required	Specifications Offered	Notes, remarks, ref to documentation	Evaluation Committee's notes
2.1.	**2**	**Servers** Four (4) quad-core CPUs, at least 2.5GHz CPU clock, x64 architecture, support for Intel-V or AMD-VT and ability to extend to 4 8-core processors 128GB RAM 1033 MHz front side bus speed Virtualization enabled 2x 146GB 15k rpm SAS or SSD for hypervisor installation Four 1Gbps Ethernet ports Two fiber channel 8Gbps ports Integrated light-out management with graphical console Case Rack mount 19" with rails Server OS according to proposed central point services Calculate OS licensing by capacity of required virtual servers for proposed solution.			
2.2.	**1**	**Storage system** External storage system rack-mount Total HDD data capacity (after RAID) minimum 5TB delivered with FC disks RAID capabilities (RAID 1+0) Active-active redundant controllers HDD hot-swapping Interface Fiber Channel SAN, iSCSI SAN, or NAS Support for SATA and/or FC and/or SAS disks Case Rack mount 19" with rails Centralized maintenance console Virtualization supported Unlimited number of hosts supported without additional licenses Unlimited capacity (only limited by architecture itself) Interconnected with 2 servers			

Item No.	Quantity	Specifications Required	Specifications Offered	Notes, remarks, ref to documentation	Evaluation Committee's notes
2.3.	**2**	**Network devices** Certified to be compatible with Virtualization SW • a Network Distributed Switch for virtualization with support for Virtualization SW ESX and ESXi hypervisors or equivalent and integration with Virtualization SW vCenter Server or equivalent. Maximum Supported Configurations: • 64 Virtualization SW hosts per VSM • 2048 virtual Ethernet ports per Virtualization SW vDS, with 216 virtual Ethernet ports per physical host • 512 active VLANs • 32 physical NICs per physical host • 256 PortChannels per Virtualization SW vDS, with 8 PortChannels per physical host Security • Ingress and egress and extended L2, L3 and L4 ACLs on Ethernet and virtual Ethernet ports with packet counters. • Virtual Service Domain for Layer 4 through 7 services virtual machine High Availability • Nonstop Forwarding: Continued forwarding despite loss of communication between the VSM and VEM. • Process Survivability: Critical processes run independently for ease of isolation, fault containment, and upgrading. Processes can restart independently in milliseconds without losing state information, affecting data forwarding, or affecting adjacent devices or services. Management • VSM installation wizard for Virtualization and network administrators • Layer 2 and 3 connectivity between VSM and VEM • OS Software CLI console • Enhanced Remote SPAN (ERSPAN) Type III: Remote port mirroring • SSH v2 • TACACS+ • Syslog – • Role based access control (RBAC) IP Multicast RFC 3376:IGMPv3 snooping Quality of Service RFC 2474:DSCP marking RFC 2698:Two Rate Three Color Marker			
2.4.	**1**	**Rack case** Server Rack 19" 42U, in set; metal door with lock, fans kit with thermostat, socket strip. Monitor 19", keyboard with touchpad or mouse.			

Item No.	Quantity	Specifications Required	Specifications Offered	Notes, remarks, ref to documentation	Evaluation Committee's notes
2.5.	**2**	**Firewall type A Central point** Firewall: NAT, PAT, Transparent (Bridge), Routing Mode (RIP v1 & v2, OSPF, BGP, & Multicast), Policy-Based NAT, VLAN Tagging (802.1Q), User Group-Based Authentication, SIP/H.323 NAT Traversal, Minimal Firewall Throughput 15Gbps, Minimum IPSec VPN Throughput 5 Gbps, Total 10/100/1000Mbps interface or 10/100/1000/10000Mbps in any combination – min 15, Virtual Private Network (VPN): IPSec and SSL, Dedicated Tunnels, DES, 3DES, and AES Encryption Support, SHA-1/MD5 Authentication, Hub and Spoke VPN Support, IKE Certificate Authentication, IPSec NAT Traversal, Dead Peer Detection, Traffic Shaping: Policy-based Traffic Shaping, Differentiated Services (DiffServ) Support, Guarantee/Max/Priority Bandwidth, Application traffic shaping , Networking/Routing: Multiple WAN Link Support, PPPoE Support, DHCP Client/Server, Policy-Based Routing, Dynamic Routing (RIP v1 & v2, OSPF, BGP, & Multicast), Management/Administration Options: Console Interface (RS-232), WebUI (HTTP/HTTPS), Telnet / Secure Command Shell (SSH), Role-Based Administration, Multi-language Support, Multiple Administrators and User Levels, Upgrades and Changes Via TFTP and WebUI, System Software Rollback, Central Management, High Availability (HA): Active-Active, Active-Passive, Stateful Failover (FW and VPN), Device Failure Detection and Notification, Link Status Monitor, Link failover.			

Item No.	Quantity	Specifications Required	Specifications Offered	Notes, remarks, ref to documentation	Evaluation Committee's notes
2.6.	**7**	**Firewall @data provider premises** Firewall , Firewall: NAT, PAT, Transparent (Bridge), Routing Mode (RIP v1 & v2, OSPF, BGP, & Multicast), Policy-Based NAT, VLAN Tagging (802.1Q), User Group-Based Authentication, SIP/H.323 NAT Traversal, Minimal Firewall Throughput 1Gbps, Minimum IPSec VPN Throughput 70Mbps, Total 10/100/1000 interface – min 5, Virtual Private Network (VPN): IPSec and SSL, Dedicated Tunnels, DES, 3DES, and AES Encryption Support, SHA-1/MD5 Authentication, Hub and Spoke VPN Support, IKE Certificate Authentication, IPSec NAT Traversal, Dead Peer Detection, Traffic Shaping: Policy-based Traffic Shaping, Differentiated Services (DiffServ) Support, Guarantee/Max/Priority Bandwidth, Application traffic shaping, Networking/Routing: Multiple WAN Link Support, PPPoE Support, DHCP Client/Server, Policy-Based Routing, Dynamic Routing (RIP v1 & v2, OSPF, BGP, & Multicast), Management/Administration Options: Console Interface (RS-232), WebUI (HTTP/HTTPS), Telnet / Secure Command Shell (SSH), Role-Based Administration, Multi-language Support, Multiple Administrators and User Levels, Upgrades and Changes Via TFTP and WebUI, System Software Rollback, Central Management High Availability (HA): Active-Active, Active-Passive, Stateful Failover (FW and VPN), Device Failure Detection and Notification, Link Status Monitor, Link failover.			

Item No.	Quantity	Specifications Required	Specifications Offered	Notes, remarks, ref to documentation	Evaluation Committee's notes
2.7.	**8**	**Personal computers** CPU: 1 quad-core processor at least 2.93GHz RAM: 4 GB PC3-10600 DDR3 SDRAM 1333MHz (1 DIMM), 3 memory slots free min, 16GB DDR3 max Hard disk: 320GB at least 8MB Cache, Serial ATA II, 7200rpm 3,5" Optical device: DVD – Recordable Dual Layer, SATA, 5,25" Expansion: 1x PCI-E 16X FH, 1x PCI-E 1X FH, 2x PCI 2.3 FH Interfaces: Min. 8x USB2.0 Graphics card: HD Graphics features, HDCP, DirectX10 or equivalent Audio: HD Audio, integrated speaker Power supply: Min 280W autosensing, 85% PSU Networking: Ethernet 10/100/1000, WakeOnLan Keyboard: Standard BH, USB, same manufacture as computer Mouse: Optical same manufacture as computer Case: Tower case black, change hard disk without tools Security and Software: Integrated security chip TCG standard 1.2, Possibility to disable all USB ports, serial and eSATA using BIOS Personal computer operating system according to proposed central point services and related configuration tools, Monitor: Min. 22" Monitor, TFT LCD, 1680 x 1050, Analog and DVI-D, Tilt Stand, Lift, Contrast 1000:1, response 5ms, lightning 250cd/m2, energy consumption 21 typical/22W max			

Appendix I: **Addresses where the Access points and the Central Service Point has to be delivered**

Item No.	Quantity	Specifications Required	Delivery address
1.1.	**1**	**Central Point Service (Enterprise Service Bus and WS Repository)**	DPC
1.2.	**7**	**Elementary Web Services**	DPC, BP, SIPA, FPA, RSMUP, DCBPA, HJPC
1.3.	**1**	**Server Virtualization**	DPC
2.1.	**2**	**Servers**	DPC
2.2.	**1**	**Storage system**	DPC
2.3.	**2**	**Network devices**	DPC
2.4.	**1**	**Rack case**	DPC
2.5.	**2**	**Firewall type A Central point**	DPC
2.6.	**7**	**Firewall @data provider premises**	DPC, BP, SIPA, FPA, RSMUP, DCBPA, HJPC
2.7.	**8**	**Personal computers**	DPC

INSTITUTION	ADDRESS
The Directorate for coordination of police authorities of B&H, responsible for the accommodation of the central services **(DPC)**	Trg BiH 1, 71 000 Sarajevo, B&H (temporary location at NOC II network center of the Ministry of Security B&H)
B&H Border Police **(BP)**	Reufa Muhića 2A, 71000 Sarajevo, B&H
B&H State Investigation and Protection Agency **(SIPA)**	Nikole Tesle bb, 71123 Istočno Sarajevo, RS, B&H

Federation Police Administration **(FPA)**	Mehmeda Spahe 7, 71000 Sarajevo, B&H
Cantonal Ministries of the Interior (using a common access point to the FPA)	Mehmeda Spahe 7, 71000 Sarajevo, B&H
RS Ministry of the Interior **(RSMUP)**	Bulevar Desanke Maksimović 4, 78000 Banja Luka, RS, B&H
B&H District Brcko Police Administration (**DCBPA)**	Trg Mladih 10, 76101 Brčko Distrikt, B&H
B&H High Judicial and Prosecutorial Council **(HJPC)**	Kraljice Jelene 88, 71000 Sarajevo, B&H