Government Service Bus (GSB)

GSB Integration Requirements

The Government Service Bus (GSB) is the central enabling set of components of the e-Government infrastructure that is based on Service Oriented Architecture (SOA). The GSB provides an infrastructure that removes any direct connection between service consumers and providers. Consumers connect to the bus and not the provider that actually implements the service. This provides location independence to all services.

The GSB also implements further value add Infrastructure or "Fabric" services. For example, security, transaction, scalability, directory, registry and delivery assurance are implemented centrally within the bus instead of having these buried within the applications or at the government agency back-ends.

The GSB architecture enables governmental entities to connect and use ready-made components of the e-Government. The diagram below shows the conceptual architecture of the GSB.

IBM WebSphere Data Power SOA Appliances are purpose-built, easy-to-deploy network devices that simplify, secure, and accelerate your XML and Web services deployments while extending your SOA infrastructure. Data Power provides configuration-based approach to meet MODEE's edge ESB requirements. The DataPower Appliance provides many core functions to applications, such as service-level management, routing, data and policy transformations, policy enforcement, access control, and hardened security—all in a single "drop-in" device.

For MODEE, Data Power provides the following key benefits.

- Platform for Vertical e-Services integration: Web services from different government entities (service providers) can be securely exposed using Data Power.
- Cross Organizational e-Services Platform: Data Power provides role-based access control to ensure the right level of secure access for cross-organizational e-Services.
- Composite e-Services integration platform: Data Power is the service composition layer that exposes composite services to service consumers.
- Shared e-Services integration platform: Data Power supports modular service integration architecture.

When deploying this IBM appliance in your network, you secure your enterprise at the Application Layer vs. at the Network Layer. DataPower is a next-generation appliance that operates on MESSAGES instead of PACKETS. This enables offloading security checks and structural checks from the service providers, there by simplifying integration while minimizing performance degradation.

Solution Benefits

Using IBM Data Power as the ESB appliance, this provides the following benefits:

- Ease of implementing security and web services in a purpose-built appliance resulting in reduced Development Lifecycle and implementation costs.
- Configuration, rather than coding: This approach offers faster time to market compared to traditional coding approaches for service integration.
- Offloading tedious security tasks from Sevice Providers (Government entities), preventing potential performance degradation
- Appliance approach provides greater security compared to software based solutions (removes periodic operating system patches, OS vulnerabilities, virtualization layer vulnerabilities, regular software patches, etc.)
- Purpose built firmware, offering wire-speed processing.
- Prepare your environment for the future: DataPower is ready for mobile and web 2.0
- Extensible architecture: add-on modules can be turned on as required.
- Highly fault tolerant device (multiple power supplies, multiple network ports) with in-built load balancing & clustering options.

The Data Power Appliance is purpose-built, easy to consume and easy to use. Data Power delivers security, common message transformation, integration, and routing functions in a network device. IBM approach helps you to leverage and scale your existing infrastructure investments.

Solution components and features

The below sections lists the used components and the utilized features within the Data Power appliance during the implementation of the Edge ESG to help meet MoDEE requirements:

- **Logging**

  IBM Data Power appliance offers a bunch of different options when it comes to logging. MODEE's main concerns when it came to logging were:

  - <u>The ability to troubleshoot a problem when one arises:</u> As for this point in the solution IBM Data Power offers a feature called 'debug probe', this feature can be enabled to log the messages temporarily and then view them at each stage within the policy execution, this also offers information like the requested and source URL/IP which should be sufficient when a problem arises at the message level.

  - <u>Being able to view and track events as they occur (mostly errors):</u> As for this DataPower's out of the box logging behavior should suffice, it offers the ability to filter the logs based on the component from which they originated and the ability to increase and decrease the level of logging details based on the current need.

  - <u>DataPower auditing</u>: Out of the box, DataPower offers the ability to log any administrational actions, by which user where they performed and when (this also included some lower level relevant action logging).

- **Security using SSL certificates**

  When it comes to SSL, the solution includes two different implementations:

  - Standard SSL over HTTP (for G2G services)

    In this scenario DataPower is issued a certificate which the service consumers should trust and accordingly be able to authenticate DataPower boxes and perform transport layer encryption. As for between DataPower and the service providers, DataPower should receive a copy of the public certificate of the entities it will connect to in order to trust them.

  - SSL with mutual authentication (for G2B services)

    As for this scenario the communication with the backend services is still done in the same manner but the communication with the consumers is done differently. In this case the first part still stands true where DataPower is still issued a certificate which the service consumers should trust but the difference is that the service consumers themselves should also be issued certificates which the DataPower should receive (public certificates) in order to perform a mutually authenticated connection.

    Mutual authentication or **two-way authentication** (sometimes written as 2WAY authentication) refers to two parties authenticating each other at the same time. In technology terms, it refers to a client or user authenticating themselves to a server and that server authenticating itself to the user in such a way that both parties are assured of the others' identity. As for the certificates issuing three different options were discussed:

  - Purchasing internationally trusted certificates
  - Using the new Jordan PKI to issue new certificates (in the future)
  - Using self-signed certificates (this option will not be used)

    DataPower supports four different formats when it comes to certificates and key:

    - DER
    - PEM
    - PKCS #8
    - PKCS #12

  Note: DataPower offers notifications for the box administrators/developers when an SSL certificate is going to expire within a month to insure minimized service downtime and a minimal impact of this event.

- **Web services proxy**

A ‘Web Service Proxy’ provides security and abstraction for remote web services. It is the object where most of the implementation will be performed and where the majority of the other features are contained. A Web Service Proxy makes it easier to implement certain features for web services based on a WSDL file.
The first step of implementing a web service in DataPower is always obtaining the WSDL (by uploading to the device or fetching from WSRR), after doing so the Web Service Proxy starts offering options starting with specifying the end point to be exposed and the protocol to be used. After that one can start applying the required policy. In the current scenario we have two policies to be applied per service the first (client to server) at the service level and another policy to apply on the way back but on a lower level and that is the operation level.
On the client to server policy:

- Within the AAA action the service credentials will be extracted from the message (Password-carrying UsernameToken element from WS-Security header), this identity will be validated against LDAP to decide whether the consumer is eligible to consume the service based on whether the identity is a member of the service group or not.
- At this stage the SLA is enforced.
- An attribute containing the identity’s access level to the services is queried and stored in context variables.
- The identity within the message is replaced with another identity which is meant to authenticate DataPower boxes at the service provider’s side.
- The destination URL is replaced with the actual service provider’s URL instead the one that came with the message here.

On the way back (server to client) each response to a consumer is filtered based on the consumer’s access level to a service using a transformation action (an XSLT style sheet) and finally the response is returned to the consumer here.Guidelines for web service integration

Government to Government - SGN

The below is a list containing all the guidelines for a service provider willing to expose a service or a service consumer willing to integrate with the GSB:

1- Messages should comply with the **XML** + **SOAP** standards.
2- All the currently implemented services follow the **SOAP** standard **version 1.1.**
3- The SOAP header must contain a **Password-carrying Username Token** element from WS-Security header.

4- The currently followed approach mandates that the **Username Token should not be signed.**
5- The SOAP message should not be encrypted nor signed.
6- The current followed approach mandates **not using Timestamp** token so that consumers with a different time or time zone settings could consume the service.
7- Both the service provider and consumer must implement and **use transport layer security**
    a. SSL version 2 should not be used
    b. SSL version 3 should not be used
    c. Weak ciphers and hashes should not be used
    d. The usage of strong ciphers only is strongly recommended
    e. It is mandatory to use TLS v1.0 , v1.1 or v1.2
    f. The usage of message compression is not recommended
    g. The usage of insecure legacy SSL should not be permitted
8- The recommended certificate format to be used is **DER encoded binary X.509 certificates** (.cer)
9- The recommended **RSA key length** for the issued and used certificates and keys is **2048.**
10- Services that can provide large chunks of data at once (ex. Search based services) are recommended to **use** some sort of **pagination** and not to return all the data at once if the result is considered large enough.
11- All the **data fields within the message body** should be marked as **optional** from the provider's side and the service consumer should be able to handle any missing or empty fields appropriately (regardless of data type).
12- The message providers are free to build the message body structure as they see fit to the service requirements as long as they comply with the relevant points mentioned above.
13- **Using any additional feature** from WS-Security or WS-Standards in general is **not recommended** unless verified and approved to be supported by the GSB.

Government to Business - Edge

In addition to all the above mentioned guidelines in the G2G section above, any entity outside the government (outside the SGN network) who would like to integrate with the GSB must comply with the below:

1- The entity must comply with the **mutual authentication** or **two-way authentication** (sometimes referred to as 2WAY authentication) specifications.
Establishing the encrypted channel using certificate-based mutual authentication involves:
    - A client requests access to a protected resource.
    - The server presents its certificate to the client.
    - The client verifies the server's certificate.
    - If successful, the client sends its certificate to the server.
    - The server verifies the client's credentials.

- If successful, the server grants access to the protected resource requested by the client.

Note: To establish this approach the entity should provide its public certificate to the GSB team (regardless of being a service provider or a service consumer) to ensure its trust as well as to receive the public certificate from GSB and insure that it is trusted from the entity's side as well.

Sample request message

```
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:u="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
1.0.xsd">

<s:Header>

    <ActivityId CorrelationId="bcf08350-0ad0-4e6a-b596-9994e137b45c"

    xmlns="http://schemas.microsoft.com/2004/09/ServiceModel/Diagnostics">9dc40624-
    0ae7-4984-8806-4e251982b213</ActivityId>

    <o:Security s:mustUnderstand="1"

    xmlns:o="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-
    secext-1.0.xsd" >

        <o:UsernameToken u:Id="uuid-1349a92e-13f7-41d1-bdde-0021a9c1d276-79">

               <o:Username>UserName</o:Username>

               <o:Password>********</o:Password>

        </o:UsernameToken>

    </o:Security>

</s:Header>

<s:Body>

    <operation xmlns="http://tempuri.org/" >

        <NationalNo>123456789</FirmNationalNo>

    </operation>

</s:Body>

</s:Envelope>
```

API Connect

IBM API Connect

“IBM API Connect is an end-to-end solution that allows users to create, secure, manage, socialize, monetize and analyze APIs. It provides a powerful set of capabilities from turning

backend RESTFUL or SOAP services into managed services. This is done by publishing APIs to API Gateways, while enforcing lifecycle and governance controls on those APIs. API Connect enables users to expose APIs, through a developer portal, targeting application developers both inside and outside their organization. Additionally, the solution’s analytics tooling helps API providers and API consumers better understand the health and consumption of deployed APIs.”

The following table explains the key steps of the API lifecycle in more detail.

Table 1. Key phases of the API lifecycle

Lifecycle Phase	**Description**
Create	Develop and write API definitions from an API development environment, eventually bundling these APIs into consumable products, and deploying them to production environments
Secure	Leverage the best-in-class API Gateway, gateway policies, and more, to manage access to your APIs and back-end systems.
Manage	Governance structures are built into the entire API lifecycle, from managing the view/edit permissions of APIs and Products being deployed, to managing what application developers can view and subscribe to when APIs are deployed.
Socialize	API Connect comes with an advanced Developer Portal that streamlines the onboarding process of application developers and can be completely customized to an organization's marketing standards.
Analyze	Developers and Product Managers alike are given the tooling in API Connect to understand their API traffic patterns, latency, consumption, and more to make data driven insights into their API initiatives.

IBM MQFT

IBM MQ Files Transfer solution is based on MQFT agents, which plays the role of either sender or receiver in case of sending files or receiving files. The same agent can be acting as both sender and receiver at the same time.

Files will be transferred across the centralized MQ infrastructure hosted in the NITC data center.

File transfers will be triggered using the following methods:

- Scheduled file transfers
- Manual file transfers
- Automatic file transfers based on monitoring a system directory

A governmental entity will be able to either send or receive files when they have the MQFT agent installed and connected from their side to the centralized MQFT environment in the NITC data center.

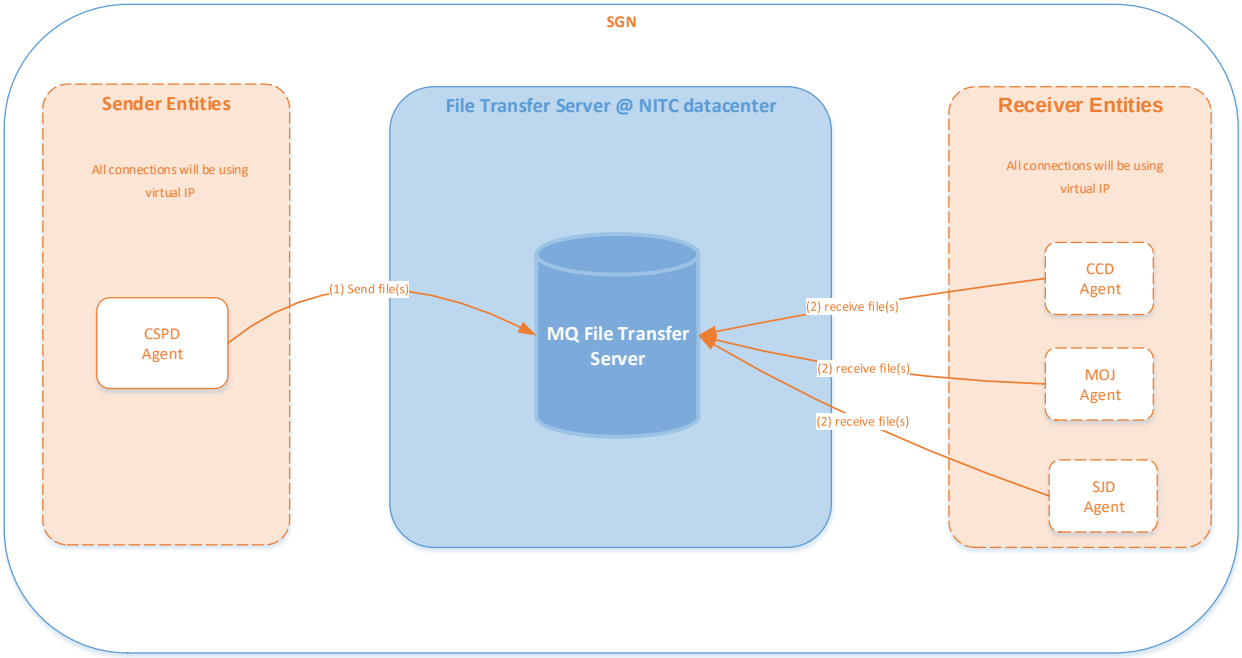

Figure (File Transfer)

IBM Publish – Subscribe

IBM MQ Publish - Subscribe solution will provide the following functionalities:

- The solution will provide one topic for each publisher
- The solution will provide a web service facility to be used for the publications
- The publisher will either use MQ APIs or the available wen service to connect to the MQ server and send publication messages. We recommend that each publisher should use MQ APIs to connect to MQ server and send the publication messages
- Publication message structure and format is considered to be the responsibility of the publisher
- It is recommended to limit the size of the publication message by using paging techniques or by sending the needed information only without extra data
- The solution will enable the system administrator to control manual subscriptions
- The administrator will create a subscription for each subscriber per topic of interest
- The solution will provide a dedicated queue for each subscriber per topic subscription
- The subscriber is responsible to connect to the MQ server hosted in the NITC using MQ APIs, and retrieve the publication messages from own queue
- Each subscriber will have access to own queue(s) only

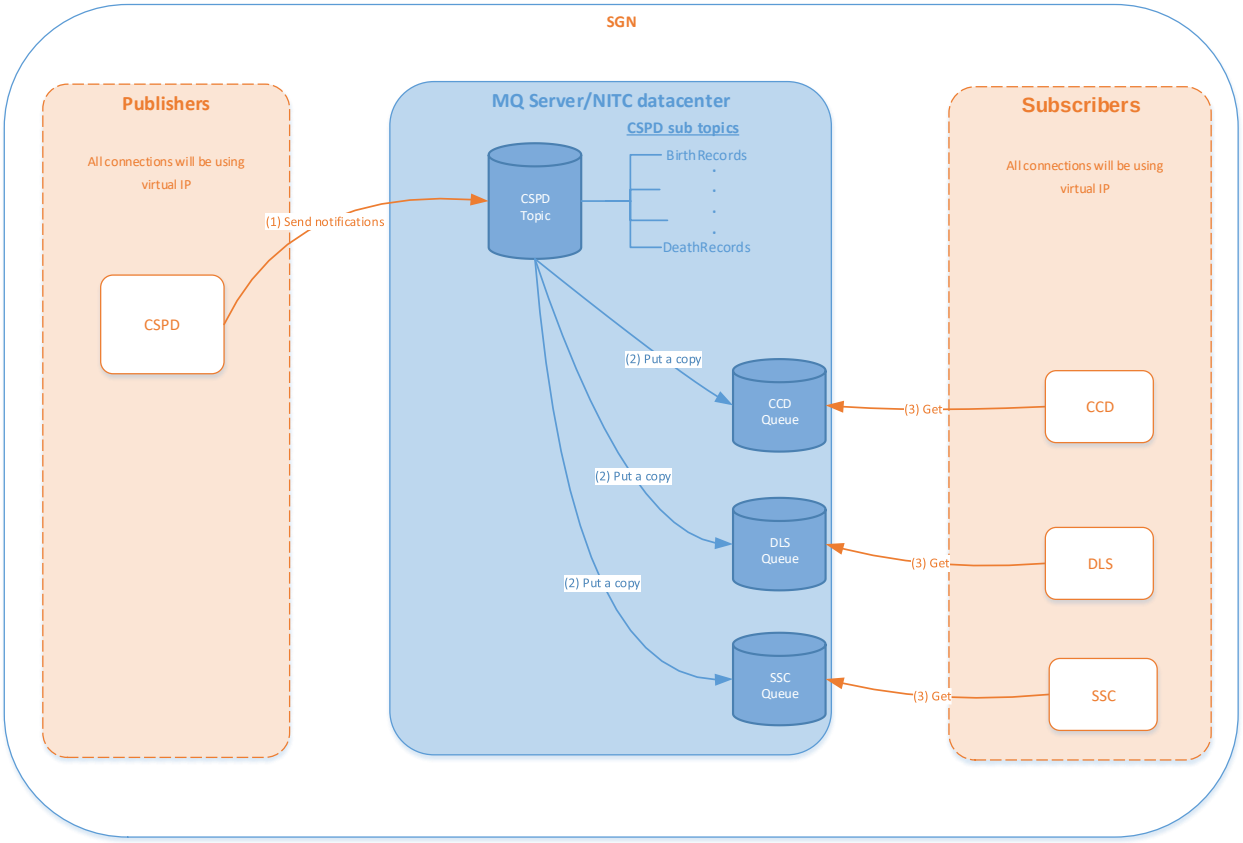

Figure (publish-subscribe Business Case)