

# Using B2B in a Dutch Government Project

**Ronald van Luttkhuizen  
Vennster  
Utrecht, The Netherlands**

## **Keywords:**

Oracle B2B, ebMS, ebXML, WUS, DigiKoppeling, architecture, SOA, case study, government

## **Introduction**

This manuscript and accompanying presentation explains a case study for a Dutch government project in which Oracle B2B is used.

Dutch government reference architectures prescribe WUS (Web Services based on SOAP and WSDL) and ebMS as standards for electronic message exchange between government organizations. OLO is a government facility that citizens and enterprises can use to request permits. OLO forwards these permit requests to applicable governments using ebMS.

The case study presents on the integration of Overijssel, a Dutch province, with OLO based on the ebMS protocol. The Oracle B2B product is used for this.

The presentation dives into the specifics of ebMS, and the capabilities of Oracle B2B. We will then present on the steps that are required to configure B2B, and how to integrate B2B with Oracle Service Bus and SOA Suite. After this presentation you will know how you can use B2B and how to integrate it with your SOA landscape.

## **The Netherlands and its Government**

The Netherlands has about 16,7 million inhabitants and an average density of 403 inhabitants per square kilometre making it a densely populated country. Its capital is Amsterdam, and the national government is seated in The Hague. The government roughly consists of:

- National level: 11 state departments, and more than 200 agencies;
- Regional level: 12 provinces;
- Local level: over 400 municipalities.

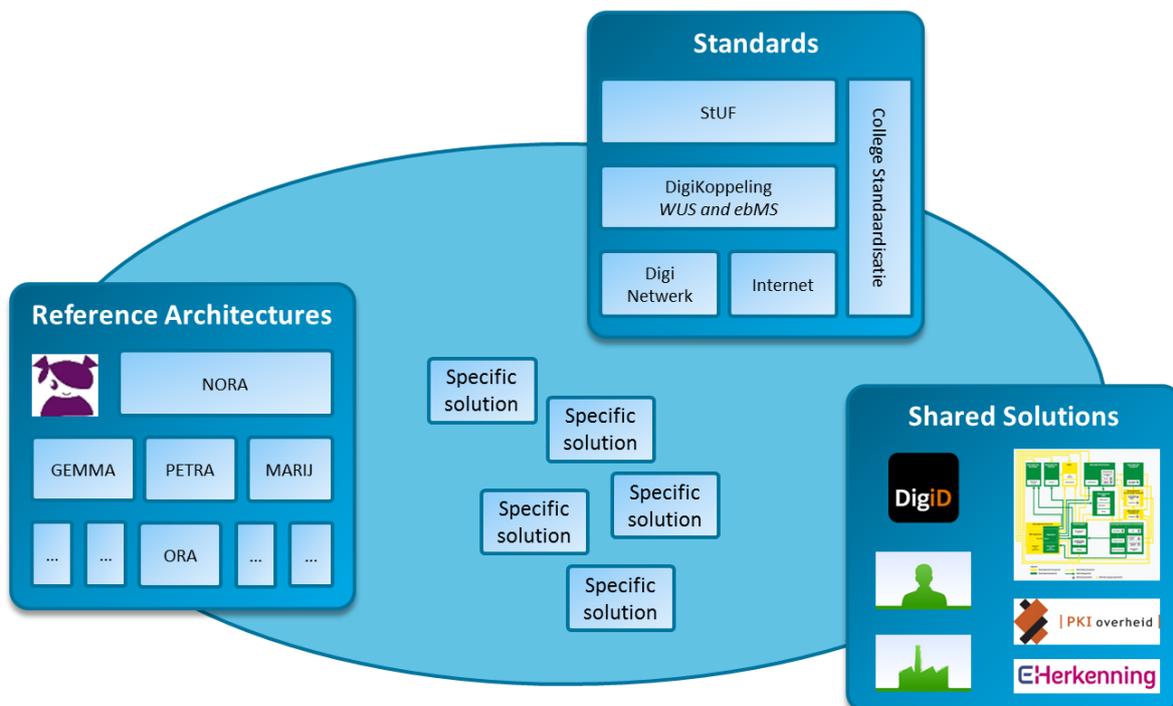
The province of Overijssel is a province located in the east of The Netherlands with about 1,1 million inhabitants (6<sup>th</sup> out of 12) and an average density of 321 inhabitants per square kilometre. Its capital is Zwolle and the province's workforce is approximately 1,000 employees.

Overijssel is tasked with regional duties and responsibilities centred around themes such as territorial planning, water management, stimulating culture, economy, and tourism, improving environment, and (financial) governance of municipalities that lie within its borders. To execute these tasks the province offers products and services such as permits, subsidies, supervision and enforcement. Recently, provinces are confronted with possible future mergers with other provinces, shifts in tasks, budget cuts due to the financial crisis, chain cooperation with other government agencies, and working together to achieve economies of scale.

## Dutch Government and IT

IT solutions of Dutch governments are influenced by the following:

- Government reference architectures. The Dutch government has created a government reference architecture called NORA that defines several guidelines such as the use of Service-Oriented Architecture (SOA) and the use of shared solutions. Most of the government sectors (municipalities, provinces, and so on) have specified a sector-specific architecture based on the NORA. An example is the PETRA: a reference architecture for provinces. Most of the government agencies also have their own specific architecture. Overijssel defined the ORA (Overijssel Reference Architecture) that is based on PETRA in turn.
- Government standards. The Dutch government strives for open-standards and open-source. The government has appointed several international IT standards as compulsory standards for government agencies. The government also defined its own standards for exchanging electronic messages, that extend international standards. These are:
  - Network level: DigiNetwerk. Private network that government agencies can use to exchange data.
  - Message exchange level: DigiKoppeling. For synchronous request/reply messages WUS (Web Services, UDDI, and SOAP) is used to exchange data. For reliable asynchronous communication ebMS (part of ebXML) is used.
  - Message payload level: StUF. Set of XSDs that specify data that is exchanged between governments such as citizens, companies, buildings, addresses, and so on.
- Shared government IT solutions or facilities. Several building blocks have been realized that can be used by all government agencies. Examples include authentication modules for Dutch citizens and enterprises, a Public Key Infrastructure (PKI) for issuing government certificates, and national databases containing authentic data on citizens, enterprises, addresses, geographic objects, and so on.



## **ebMS**

The ebMS standard is part of the ebXML (Electronic Business XML) standard and specifies how electronic XML documents are exchanged between businesses (service provider and consumer) while guaranteeing several Quality-of-Service aspects such as guaranteed delivery and security. Such Quality-of-Service aspects are optional in “plain” SOAP Web Services and REST-based services.

The ebXML standard was created as alternative to more heavy-weight standards such as EDI. The ebXML standard is specified by OASIS (Organization for Advancement of Structured Information Standards) and UN/CEFACT (United Nations Centre for Trade Facilitation and Electronic Business) and builds upon existing (Web Service) standards such as SOAP, HTTP, SSL/TLS, and WS-Security. Most of the ebXML standards focus on design-time aspects such as describing processes and business capabilities. The capabilities of a company are registered in electronic documents called Collaboration Protocol Profile (CPP).

The ebMS standard is a runtime standard for exchanging messages. To achieve the Quality-of-Service aspects (meta) information is added to the SOAP-envelope. Tools that support ebMS can generate, parse, and act upon these headers so the Quality-of-Service aspects are enforced. Formal contracts called Collaboration Protocol Agreement (CPA) need to be specified between every consumer and provider of the service. A CPA is based on the CPPs of the trading partners involved. The SOAP-headers of ebMS messages contain consumer-specific information based on these agreements making ebMS a point-to-point integration technology; i.e. a message exchange is specific to one consumer.

DigiKoppeling specifies two standards. The other standard for message exchange between Dutch governments next to ebMS is WUS. WUS is specified as SOAP-based Web Services using SSL/TLS for transport security. DigiKoppeling specifies various profiles for ebMS and WUS standards, each with their own Quality-of-Service aspects. For example, WUS profile 2W-be-SE that adds signing and message encryption based on WS-Security. Both WUS and ebMS allow for various transport protocols to be used.

The main difference between ebMS and WUS is that ebMS is aimed at point-to-point integration; every service consumer needs to define a separate contract with the service provider specifying what documents can be exchanged and what Quality-of-Service (QoS) apply. This increases the complexity of maintaining ebMS-based integrations and the complexity of message exchanges itself. WUS provides a point-to-many integration in which there are no specific contracts and consumer-specific headers per service consumer.

## **Case study**

The case study involves the realization of an automated integration between the SOA infrastructure and permits backoffice application of Overijssel and the OLO (Omgevingsloket Online); a shared government IT solution for handling permits.

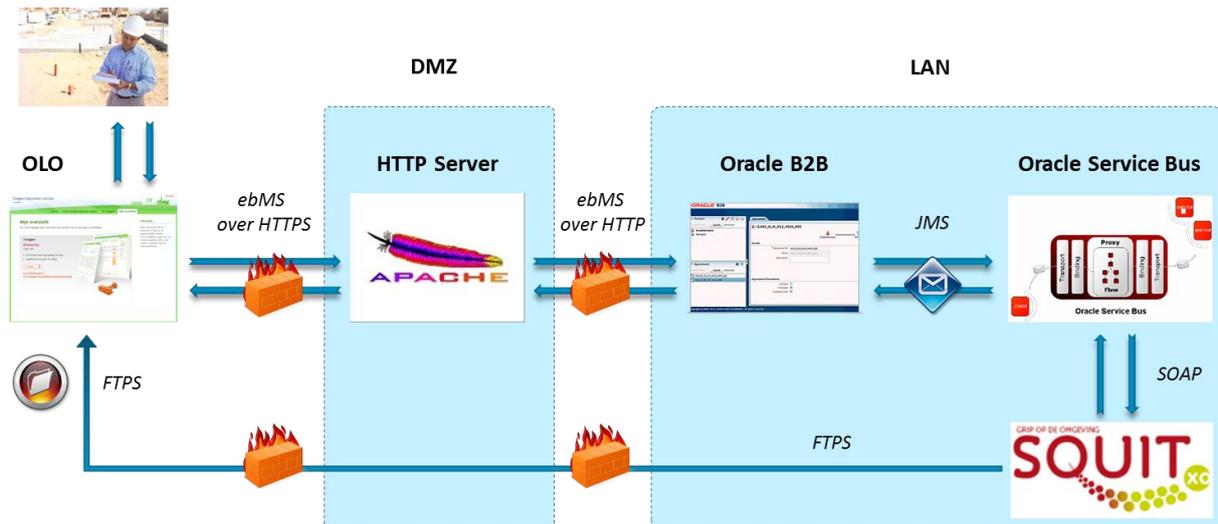
Several activities that influence the surrounding such as felling trees and building new structures require permits. Depending on the complexity and impact of the activities different government agencies need to approve different permits. In the past citizens and enterprises were forced to apply for multiple permits at multiple government agencies and coordinate this process themselves. A new law called the WABO changed this by stipulating that . Using the new shared solution OLO, citizens and

enterprises can request a permit. OLO will then handle the communication between all involved government agencies.

Government organizations can either integrate with OLO manual using a web console and email, or integrate in an automated fashion using the DigiKoppeling ebMS standard for exchanging messages.

## Solution

The solution that Overijssel implemented is depicted in the following figure:



The solution consists of the following components:

- **OLO.** The shared government solution that citizens and enterprises can use to request permits. OLO determines which government agencies should process the request and signals these agencies. OLO is then responsible for the orchestration of the permit as a whole.
- **Squit XO.** A standard backoffice application that Overijssel uses to handle permits. Squit XO exposes a SOAP Web Service and is able to consume a Squit XO
- **Oracle Service Bus.** Used to translate the different protocols and payloads between the backoffice application Squit XO that communicates based on SOAP and Oracle B2B that is integrated using JMS. Note that Oracle B2B supports various protocols. JMS was chosen since it enables asynchronous communication, Overijssel has experience with this, and B2B is relatively easy to integrate with JMS using custom transport headers.
- **Oracle B2B.** Used to setup ebMS message exchange between Overijssel and OLO using reliable messaging.
- **Apache HTTP Server.** The Apache HTTP Server, and several add-ons, are responsible for SSL-offloading enforcing SSL transport security between OLO and Overijssel. SSL is applied in a two-way fashion in which both client and server use certificates and keys for transport security.
- **Network components.** The various network components such as firewalls, network appliances, and other security and gateway components are configured so that communication between OLO and Overijssel is allowed.

## Oracle B2B

Oracle B2B 11g is a gateway product that is used to exchange documents between businesses that are based on certain industry standards (that are often not supported out-of-the-box by SOA and BPM platforms). Oracle B2B provides security, reliability)

The exchange of electronic documents involves several layers for which B2B provides support. The underlined protocols and standards are the ones that are applied in this case.

- Document types. Support for among others HL7, EDI, RosettaNet, xCBL, and Custom (XML, CSV, flat, etc.).
- Packaging of documents. Support for among others MIME, SMIME, gZIP, XMLDSig, XMLEncryp, SOAP (as part of ebMS).
- Transport of documents. Support for among others AQ, JMS, File, (s)FTP(s), HTTP(s), SMTP, IMAP, POP3, MLLP, and TCP/IP.
- Messaging & requirements. Support for among others RNIF, AS1, AS2, ebMS & digital signing, time to acknowledge, and retry.

Oracle B2B is a component of Oracle SOA Suite. Some “adapters” such as the EDI, Healthcare, RosettaNet and ebXML adapters require additional licenses.

Oracle SOA Suite provides an out-of-the-box adapter to integrate with Oracle B2B. This adapter can be used for inbound communication to start SCA composites (service), or outbound communication in which SCA composites execute a callout to Oracle B2B (reference). The adapter supports native integration based on a WSDL description (default) and asynchronous communication based on JMS or AQ.

Other tools such as Oracle Service Bus and 3<sup>rd</sup> party tools can integrate using one of the transports that Oracle B2B supports; e.g. JMS, AQ, FTP, or File. When integrating using JMS or AQ, headers are added to the messages that flow to Oracle B2B. Based on these headers, B2B can decide what document exchange to initiate.

There are several tools involved for managing Oracle B2B:

- WebLogic Server Console; a web-based console for managing the underlying infrastructure (application server), resources (JMS, JDBC, and so on), and transport-level security (SSL/TLS).
- Enterprise Manager Fusion Middleware Console; a web-based console for managing the SCA composites that interact with Oracle; either inbound or outbound.
- B2B Console; a web-based console for administrating B2B itself and managing the various document exchanges and accompanying agreements between businesses. This console also offers functionality for exporting and importing configurations between B2B runtimes.
- Next to these consoles, B2B offers a dedicated command-line interface, Java API's, and a Web Service (<http://server:port/b2b/services>) to interact with Oracle B2B.

## **Implementation**

The implementation of the integration between OLO and Overijssel consisted of the following steps:

- Start

- Acquire a unique government identifier called “Overheid Identificatie Nummer” (OIN) that provides access to the Government Service Registry.
- Use the OIN to acquire government issued PKI certificates.
- Configure Infrastructure
  - Install WebLogic Server, SOA Suite, and Oracle B2B.
  - Configure transport security (SSL/TLS) in the HTTP Server and add-ins.
  - Configure network components such as firewalls.
- Configure Oracle B2B
  - Create a Collaboration Protocol Agreement (CPA) between OLO and Overijssel using the Government Service Registry and the government CPA tool.
  - Configure Oracle B2B: import CPA, import B2B configuration, and/or use B2B Console (trading partners, document definitions, channels, agreements).
- Integrate Oracle B2B
  - Connect Oracle B2B with Oracle Service Bus using JMS.
  - Connect Oracle Service Bus with the backend system using SOAP.
  - Configure the HTTP server and add-ins, and set the endpoint in Oracle B2B.
  - Configure OLO using its Admin Console.
- Test and management
  - Validate the compliancy of Oracle B2B with the government ebMS compliancy tool.
  - Integration testing using the OLO test environment.
  - Implement management of the integration by IT-Operations.

The steps that are underlined are explained in more detail in the slides using screenshots.

## Summary

Dutch governments use reference architectures, standards, and shared solutions to create IT solutions for their own organization. One of these shared solutions is OLO, that citizens and enterprises can use to request permits. Governments can either integrate manually or automatically with OLO as part of their permits business process. For automated integration, ebMS must be used as message exchange standard. DigiKoppeling, the Dutch government standard for exchanging electronic messages, specifies the way ebMS should be used for guaranteed delivery of messages.

The province of Overijssel has implemented Oracle B2B to support ebMS exchanges. The presentation and manuscript explains what steps are required for implementation. Oracle B2B supports a variety of document exchanges based on industry standards:

- Document types. Support for among others HL7, EDI, RosettaNet, xCBL, and Custom (XML, CSV, flat, etc.).
- Packaging of documents. Support for among others MIME, SMIME, gZIP, XMLDSig, XMLEncryp, SOAP (as part of ebMS).
- Transport of documents. Support for among others AQ, JMS, File, (s)FTP(s), HTTP(s), SMTP, IMAP, POP3, MLLP, and TCP/IP.
- Messaging & requirements. Support for among others RNIF, AS1, AS2, ebMS & digital signing, time to acknowledge, and retry.

The implementation of Oracle B2B proved to be relatively easy. Some issues that occurred during the implementation were mainly because of ebMS itself, not Oracle B2B. These issues are:

- The ebMS standard is (more) complex compared to “plain” SOAP Web Services or REST-based service due to the Quality-of-Service aspects it enforces using SOAP headers.
- Maintenance of ebMS-based integrations is more heavy-weight due to point-to-point agreements that have to be created between every consumer and provider. Plain Web Services and REST-based services support point-to-many integrations.
- There is little knowledge and experience of ebMS. Implementation of ebMS often requires extra tools and extra costs. The future of the ebMS standard is uncertain.
- There are alternatives to ebMS for reliability and security, for example the use of WS-RM and WS-Security. Those alternative standards can be used alongside SOAP-based Web Services. The Dutch governments recently also adopted WS-RM as alternative to ebMS, thereby moving away from ebMS over time.

**Contact address:**

**Name**

Vennster

Postbus 31457

6503 CL, Nijmegen, the Netherlands

Phone: +31 (0)6 52456043

Email [ronald.van.luttikhuisen@vennster.nl](mailto:ronald.van.luttikhuisen@vennster.nl)

Internet: vennster.nl

Twitter: rluttikhuisen

LinkedIn: <http://nl.linkedin.com/in/soaarchitecture>