



**Corporate Services for the  
Natural Resource Sector**

**Information Management Branch**

---

## **Standards for Application Diagrams**

**Last Updated:** June 20, 2011  
**Version:** 1.0.0  
**Document:** NRS\_Standards\_for\_Application\_Diagrams.1.0.0.docx

# Table of Contents

<b>1. VERSION CONTROL .....</b>	<b>4</b>
<b>2. INTRODUCTION.....</b>	<b>5</b>
2.1. Purpose.....	5
2.2. Audience.....	5
2.3. Scope/Exclusions .....	5
2.4. Assumptions.....	5
2.5. Definitions.....	5
<b>3. TYPES OF DIAGRAMS.....</b>	<b>6</b>
3.1. Application Pattern Diagram.....	6
3.2. Application Layout Diagram.....	6
3.3. Application Blueprint Diagram .....	6
<b>4. DIAGRAMMING ENVIRONMENT .....</b>	<b>6</b>
4.1. Diagramming Tools.....	6
4.2. File Management.....	7
<b>5. NOTATION (VISUAL REPRESENTATION).....</b>	<b>8</b>
5.1. General Layout.....	8
5.2. Node (Device or Execution Environment).....	8
5.3. Component .....	9
5.4. Artefact.....	9
5.4. Associations (Links).....	9
5.5. Zones .....	10
5.6. Annotation/Notes .....	10
5.7. Stereotypes .....	12
5.8. Element Colour Fill.....	13
5.9. Additional Detail .....	13
<b>APPENDIX A: ENTERPRISE ARCHITECT NOTES .....</b>	<b>14</b>
A.1 Exporting a diagram from EA in XMI.....	14

A.2	Avoiding use of stereotype icons .....	15
A.3	Adding a note to a link .....	16
A.4	Using an existing diagram as a starting point .....	16
<b>APPENDIX B: SECURITY ENHANCEMENT PROJECT (SEP) ZONATION .....</b>		<b>17</b>
<b>APPENDIX C: DIAGRAM EXAMPLES .....</b>		<b>18</b>
C.1.	Application Pattern: Secured Java over Oracle.....	18
C.2.	Application Pattern: Lightweight Development Approach.....	18
C.3.	Application Layout: VMAD .....	19
C.4.	Application Blueprint: PSCIS in PROD (draft) .....	20
C.5.	Application Blueprint: SOFT / PROD .....	21

## 1. Version Control

Document Version	Revision	Date	Author(s)	Change Reference
	Draft	September 2010	Todd Glover	Draft for review. Minor changes September 2010 – February 2011
1.0.0	Draft	2011-03-17	Todd Glover	Intended for initial release
1.0.0	Draft	2011-06-20	Todd Glover	Incorporates Standards Committee feedback
1.0.0	Final	April 2012	L Solomon	Published

## **2. Introduction**

### **2.1. Purpose**

This document defines standards related to the tooling, representation, and storage of application diagrams. Applications diagrams visually describe the technology perspective of the construction of applications and information systems, and are types of UML deployment diagram.

Application diagrams are used in a variety of ways, and constitute certain deliverables within the Systems Development Life Cycle.

### **2.2. Audience**

This document is directed at vendors and ministry staff who create, manage, or consume application diagrams for the Ministry.

### **2.3. Scope/Exclusions**

The scope of this document covers all application diagrams delivered to or maintained by the Ministry.

Where conflicts, if any, are perceived between this document and other standards, the Ministry Architecture group must be consulted.

### **2.4. Assumptions**

It is assumed that the audience has a basic working knowledge of servers, applications and how they interact to form an information system. It is further assumed that the audience is familiar with one or more diagramming tools such as Enterprise Architect, Microsoft Visio, or Microsoft PowerPoint.

### **2.5. Definitions**

The following definitions apply throughout this document.

#### **2.5.1. Standards**

A standard is a specific statement of the rules and constraints governing the naming, contents, and operations of software. A standard must be followed. There is a contractual obligation on the part of the vendor/developer to adhere to all relevant standards.

#### **2.5.2. Guidelines**

A guideline is a method or custom, which through common usage has become an accepted method of work. A guideline is not enforced, and is not a standard.

#### **2.5.3. Alternate**

Where several choices are offered of which one must be used, the words *alternative* or *alternate* are used, with the preferred choice identified by the word *preferred*. In these latter cases any listed option is acceptable although the *preferred* choice is preferred. It remains mandatory to use one of the choices described.

#### **2.5.4. Sector, Ministry, IMB, CSNR**

The word “Sector”, when referring to an organization, refers collectively to the provincial ministries that make up the natural resources sector.

Unless otherwise specified, "Ministry" is taken to collectively mean any Ministry or organization within the British Columbia natural resource sector. Specifically, this includes any organization whose information management takes place under the authority of the Chief Information Office for the Natural Resource sector.

CSNR refers to the Corporate Services for the Natural Resource Sector, serving the province’s natural resource ministries.

IMB refers to the Information Management Branch of the CSNR.

### **3. Types of Diagrams**

This document discusses three types of diagrams. They are types of UML Deployment Diagrams. From the most abstract to the most specific, the types of application diagrams for IMB use are:

#### **3.1. Application Pattern Diagram**

This diagram shows an architectural pattern, and is concerned with the broad interactions between types of services, applications and systems. Computers and applications are abstracted. Example: [Secured Java over Oracle Application Pattern](#). These are created and managed by IMB Architecture Group.

#### **3.2. Application Layout Diagram**

This shows the nodes, components, and interactions for a specific application, but specific credentials, server names, and versions are typically abstracted (for example, a node might be “Database Server” but not shown with a specific server name). Example: [#Application Layout: VMAD](#) These diagrams are typically developed by vendors as part of application development or maintenance. They are preferred to be developed by adding detail to an application pattern diagram, or abstracting an existing application blueprint diagram.

#### **3.3. Application Blueprint Diagram**

The most detailed application diagram type, this shows the links and components for an application within a specific environment (e.g. TEST, PROD). It will typically show names of specific servers, component versions, and links may be elaborated with credentials or more descriptive dependencies. Example: [#Application Blueprint: SOFT / PROD](#). These are typically developed by the vendors as part of the application development or maintenance.

### **4. Diagramming Environment**

#### **4.1. Diagramming Tools**

The preferred tool to create and manage diagrams is Enterprise Architect, referred to as “EA” in this document. Acceptable alternatives are Microsoft Visio or Microsoft PowerPoint.

## 4.2. File Management

### 4.2.1. Diagram States

Each diagram shall exist in two states, one an editable file and one as an image for inclusion in documents. The editable form shall be the native format of the approved tool (above). The image shall be in PNG or JPG format, with PNG preferred.

- Related diagrams may be stored in the same editable file (e.g. Layout and Blueprint diagrams for a particular application; or multiple application Pattern diagrams).
- Enterprise Architect: Each diagram (or related group of diagrams) shall be stored as an XMI export. See appendix for instructions for exporting diagrams in XMI.
- PowerPoint, Visio: save in native format.
- An image of the diagram in PNG format suitable for printing, including in documents, etc.

Application layouts & blueprint files are typically stored in the application project folder in the same directory as the design document. Application pattern diagrams are typically stored in an area related to overall standards.

### 4.2.2. Bundled Images

For the editable files, diagrams may be bundled (i.e. multiple diagrams in one file) in this way:

- Patterns may be bundled together into a file.
- Layouts and Blueprints may be bundled together on a per-application basis (i.e. all the layout and blueprints for application XYZ may be bundled together into one file).

File naming (below) must make clear the extent of the file's contents.

### 4.2.3. File Naming

File names shall conform to the IMB Standards for Versioned Documents, extended with the following:

- Enterprise Architect files shall end with file extension “.eap”
- Visio files shall end with file extension “.vsd”

The ‘type’ of files (as referred to in the Standards for Versioned Documents) shall be:

Diagram type	Name ‘part’	Example filename
Application Pattern	Pattern	patterns.secured.1.0.0.eap (for a bundle including a number of patterns related to ‘secured’); pattern.secured_java_over_oracle.1.0.0.pptx

Application Layout	Layout	MyApp.2.6.3.layout.vsd; MyApp.2.6.3.application_diagrams.eap (where all the diagrams for MyApp are bundled together).
Application Blueprint	Blueprint	MyApp.2.6.3.blueprint.pptx

## 5. Notation (Visual Representation)

Notation is based on a visual standard. When working in EA, specific ways of instantiating the visual elements may be defined.

### 5.1. General Layout

Preferred layout for elements is to have client elements (e.g. user PCs) toward the left and back-end services (e.g. database) toward the right. This is a guideline and should defer to the need to keep the diagram legible and to convey its meaning clearly.

Colours shall be chosen such that the diagram is legible whether printed in black and white or in colour.

Each diagram must include a title block which is a box containing the diagram name, title, date and time the diagram was last modified, the author, and the application system name as recorded in the Application Repository.

In enterprise architect, diagrams are of type “Deployment Diagram”.

Elements that are important in understanding an application are to be included; others may be assumed. For example, SAN storage may typically be assumed (i.e. not depicted), unless there is something about use of SAN for a given pattern or application that makes depiction of SAN helpful. For human-facing systems, the User PC and its components should be included.

Similarly, when considering whether to make an element a node or a component, consider clarity and the degree of detail that is helpful to understanding the system. For example, a web browser would typically be a component; however, if configuration and deployment takes place in the browser, then it could prove useful to depict the browser as an execution environment, containing program components, JavaScript execution environment, or configuration artefacts. There is no hard & fast rule for this, and the diagrammer must make choices based on experience as well as taking guidance from the Ministry.

### 5.2. Node (Device or Execution Environment)

Nodes include both devices (a physical device offering computational resources) or an execution environment (which offer an execution environment for components that are deployed on it). It may contain nodes, components, and artefacts.

Preferred depiction is a 3D box. Alternate depiction is a rectangle. Must be stereotyped. Typical stereotypes are:

- For device nodes: «user pc», «server», «unix server»



- For execution environment nodes: «OS», «workflow engine», «database», «J2EE container», «filesystem», «execution environment».

A node may be the source or target for a communication link or a dependency.

### 5.3. Component

A component is an element that represents a service, and may contain components and artefacts.

Preferred depiction for a component is a rectangle containing a component icon (consisting of a rectangle with two nested icons) or a rectangle with two rectangles nested on its left edge (see example). Alternate depiction is a rectangle. Components should be stereotyped where it adds meaning. A component may be the source or target for a dependency. Typical component stereotypes include schema, java. An example component that might not be stereotyped is “Apache” where it is not treated as an execution environment, or “Siteminder Agent”.

### 5.4. Artefact

An artefact is an element of configuration within a system. Examples may include: configuration file; rows within a shared authorization schema; a custom siteminder realm.

Preferred depiction is a rectangle with a stereotype of “artifact”. (Enterprise Architect uses US spelling.)

### 5.4. Associations (Links)

Lines between elements (called “Links” in Enterprise Architect) are used to show a communication association (e.g. a network link) or to show a run-time dependency association between elements.

Depiction standard:

- **Communication relationship** (or “connection”): a solid line without arrows, linking nodes (not components). Must be stereotyped, usually with the connection protocol (e.g. <<sql\*net>>, <<http>>, <<https>>). Cardinality may be indicated if it adds meaning, using conventional UML: “\*”, “1”, “1..\*”, etc.
- **Dependency relationship**: a dashed arrow with an open arrowhead, optionally stereotyped with “<<use>>”, linking nodes/components. The arrow goes from the caller to the called element (for example, a client calls a server, and an application may open communication (call) to a database). A two-headed arrow denotes that either element may invoke the other (e.g. in a peer-to-peer network).
- **Other runtime associations**: It may be desirable to extend the diagram, especially the blueprint diagram, with additional runtime associations. Where used, these *must* be stereotyped, and *must* be represented as dashed arrows with an open arrowhead. An example might be an information flow, stereotyped with “<<flow>>”.

Additional information related to paths may be added using notes or additional lines of text associated with the link, including such things as communication protocol, credentials, read-only/read-write, firewall filters, ports, etc.

## 5.5. Zones

Depiction of network zones is optional. One or more network zones may be used where helpful, or as dictated by IMB. A network zone is shown by a solid-coloured background surrounded by either no border or a dashed border.

Where zones are used, colours and definitions are based on the OCIO SEP Segmentation Model, extended by IMB's existing client-managed server subnet. Zones should be labelled with stereotypes.

Zone name	Colour	
	RGB	Example
<<internet>> Internet zone	255/255/153	Pale yellow
Span/BC (included for information only – not used in IMB diagrams)	255/204/153	Pale orange
<<dmz>> DMZ	255/153/204	Pink
Extranet (included for information only – not used in IMB diagrams)	204/255/255	Pale blue
<<zone c>> Zone C (workstation zone)	204/255/204	Pale green
<<zone b>> Zone B (Application zone, including existing IMB client-managed subnet). Alternative (legacy) is <<imb server zone>>	153/204/255	Medium blue
<<zone a>> Zone A (Highest security database zone)	204/153/255	Violet



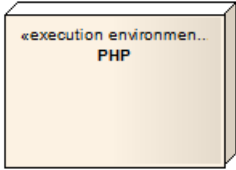


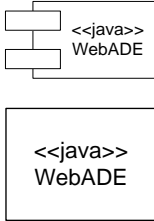
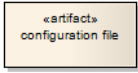
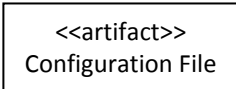
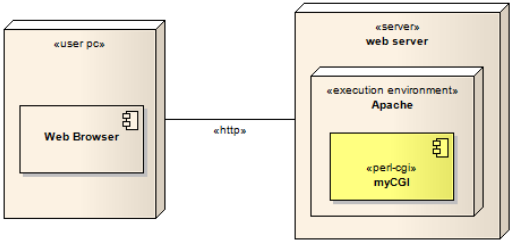
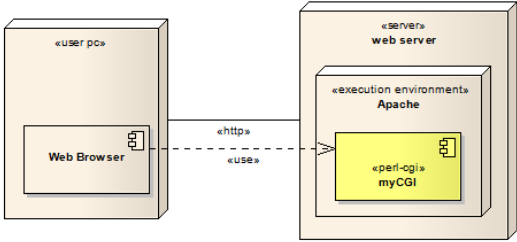
It is preferred that zones, where used, are laid out with the zones that are higher in the list toward the left of the diagram and the zones lower in the list to the right, preserving the overall diagram trend of client services on the left and back-end services on the right.

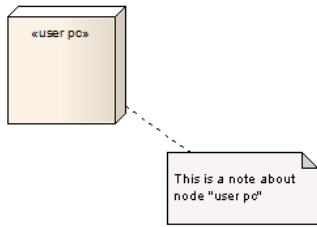
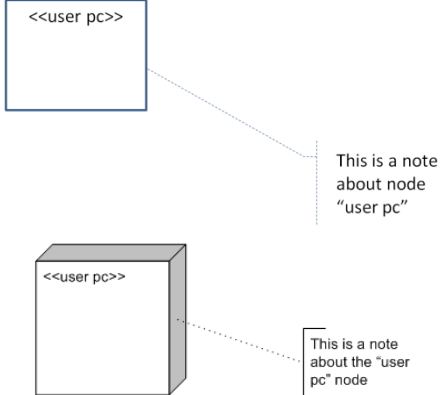
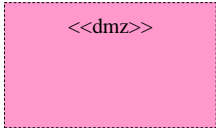
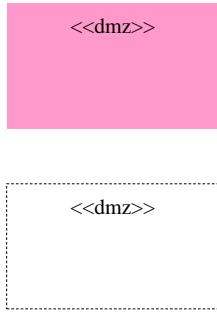
An alternative zonation scheme is to use zones to show different parts of the system's organizational responsibility: so IMB servers may be in one zone, ILMB's and other shared service providers in another, etc. This scheme is acceptable, and if used the visual standards are:

- The colours used for the zones shall allow the diagram to remain legible whether printed in black and white or in colour;
- Each zone will be labelled, e.g. "<<shared services>>".


## 5.6. Annotation/Notes

May be attached to nodes, elements, artefacts, or links, or may float free. Preferred depiction is a dog-eared box without stereotype, attached to its associated element (if any) using a dotted line. Alternate depiction is to use round -cornered boxes, or borderless callouts.

Element	Preferred Representation	Alternate Representation(s)
Device Node		
Execution Environment Node		
Component		
Artefact		
Communication relationship (connection) link		<p data-bbox="1036 1197 1380 1226">A solid line without arrowheads</p>
Dependency (runtime association) link – shown in addition to communication relationship		<p data-bbox="1029 1470 1386 1533">A dashed line with arrowhead; arrowhead preferred to be open</p>

Element	Preferred Representation	Alternate Representation(s)
Note – shown attached to a node		
Network zone		

**5.7. Stereotypes**

Stereotypes shall be used, and be depicted using text rather than icons. (E.g. “<<user pc>>” rather than the Enterprise Architect icon “”.) See Appendix A for notes on avoiding use of stereotype icons in Enterprise Architect.

Preferred (typical) stereotypes are listed here; others may be used as necessary.

Element type	Typical stereotypes
Nodes & Components	<<server>>, <<user pc>>, <<database>>, <<filesystem>>, <<execution environment>>
Components	<<schema>>, <<java>>
Artefacts	<<artifact>>
Network Zones	<<internet>>, <<dmz>>, <<zone c>>, <<zone b>>, <<imb server subnet>>, <<zone a>>
Dependency association link (- - - - - - ->)	Blank, or <<use>>
Connection link (-----)	<<https>> , <<http>>, <<sql*net>>, <<scp>>, <<ftp>>

## **5.8. Element Colour Fill**

It is preferred that in Layout and Blueprint diagrams, that nodes, components, and artefacts that are unique to the application under consideration are filled in a lighter colour than the rest of the elements. It is preferred that the colour distinction remain whether the diagram is printed in colour or in black and white.

Recommended colours:

- General elements: EA default.
- Application-specific elements: light yellow

## **5.9. Additional Detail**

Additional diagram elements, such as ports and interfaces, may be modelled where helpful, especially in the application blueprint diagram.

For example, it may be useful to model a dependency to a specific interface of a component where that component exposes multiple interfaces. In this case, the interfaces would be modelled and the dependencies would be directed toward one or another of the component's interfaces.

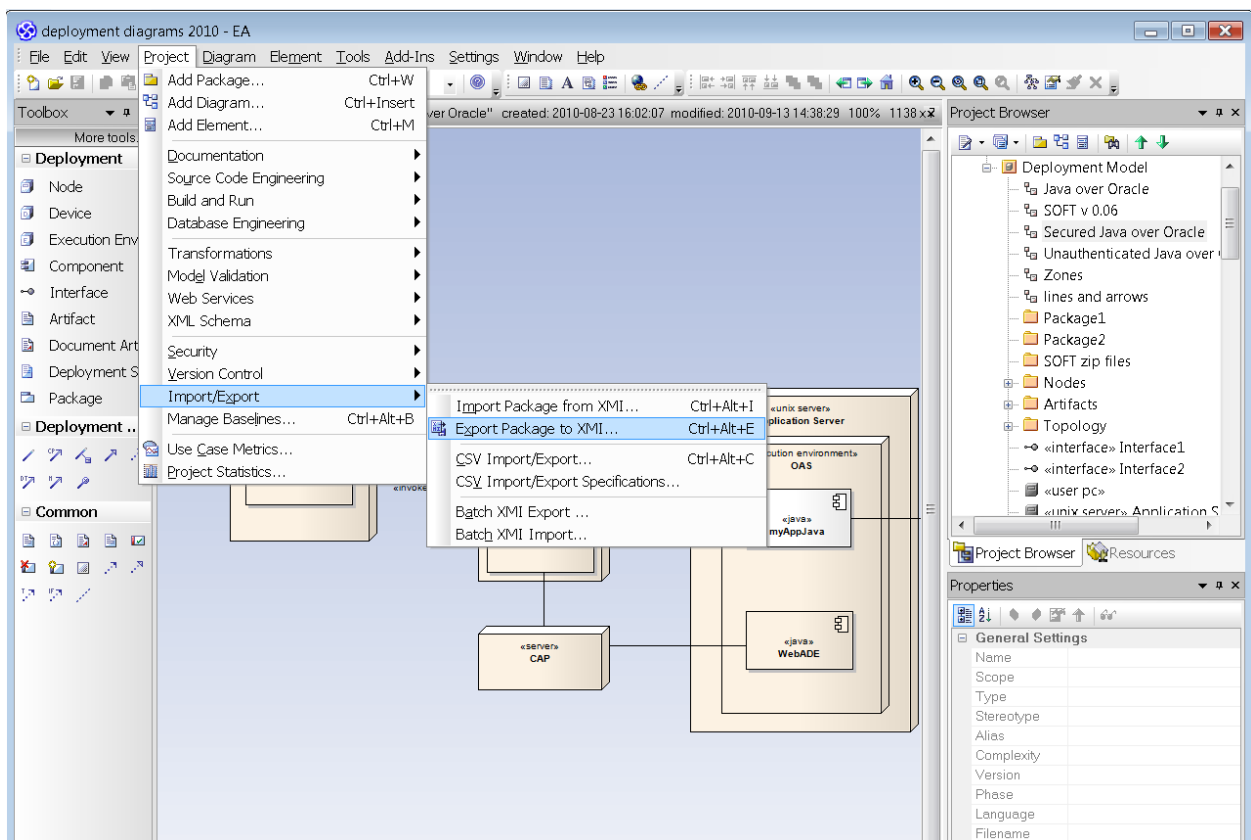
Another example is the use of additional detail concerning data flows, periodicity or servers' geographic location where this is necessary to understand the architectural structure of the application.

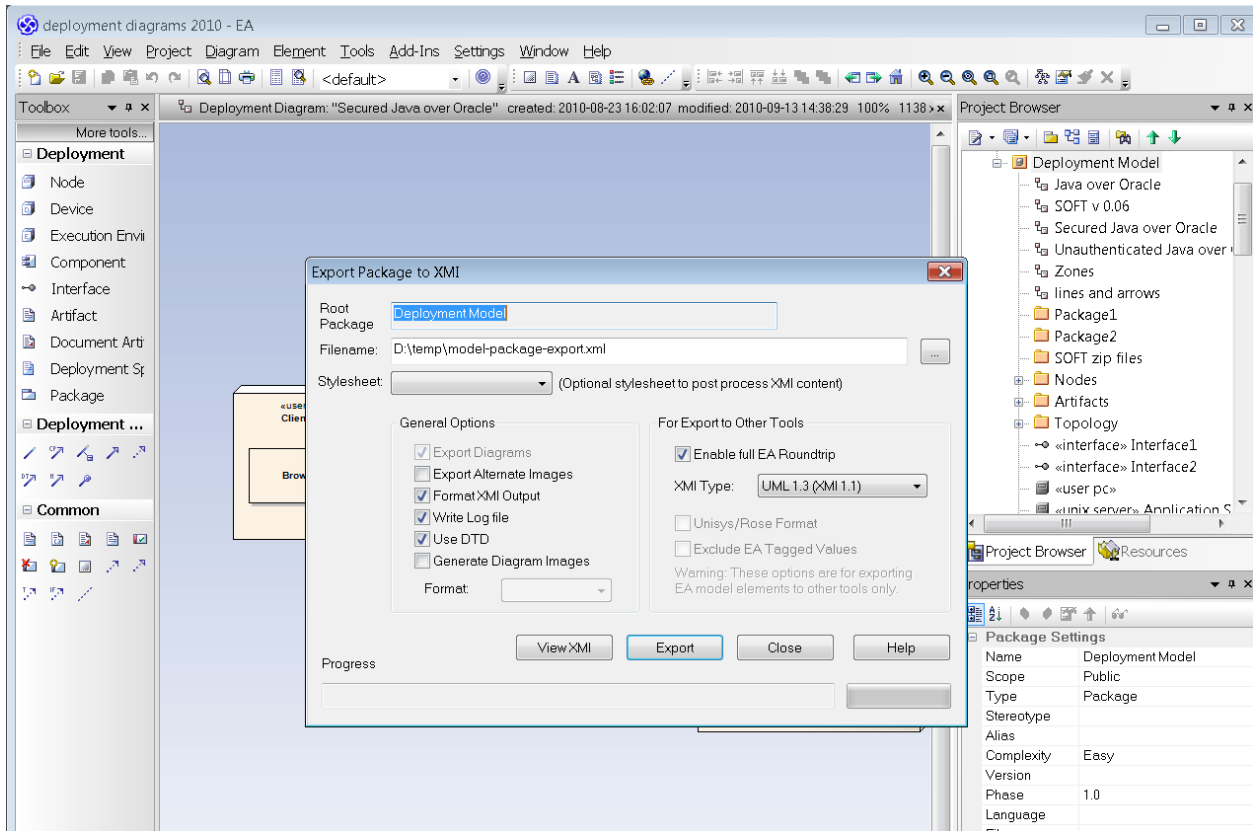
# Appendix A: Enterprise Architect notes

## A.1 Exporting a diagram from EA in XMI

XMI files should have the extension .XML.

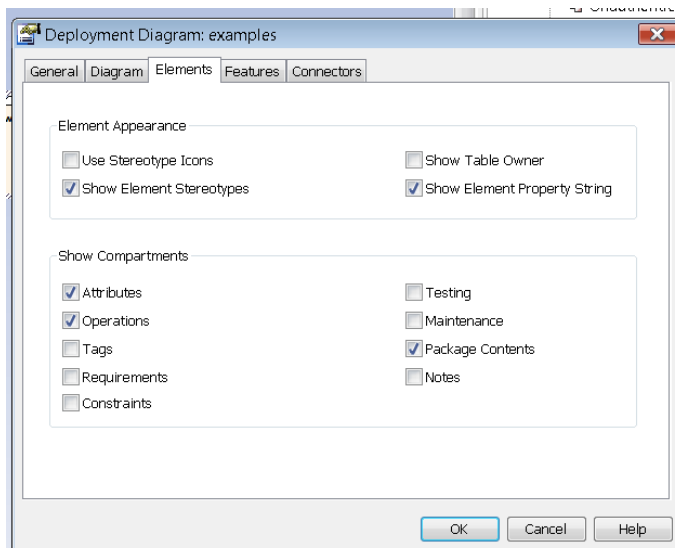
- Select the chosen diagram in the RH panel
- Menu-navigate to Project->Import/Export->Export Package to XMI
- Enter appropriate filename (\*.xml)
- Make sure “round trip” is selected
- “Export”





## A.2 Avoiding use of stereotype icons

Select Diagram->Properties, and on the “Elements” tab, un-check the “Use Stereotype Icons” checkbox.



### ***A.3 Adding a note to a link***

It can be non-obvious to add a note to a link.

- Select the link
- Right-click for context menu; select “Attach note or constraint”

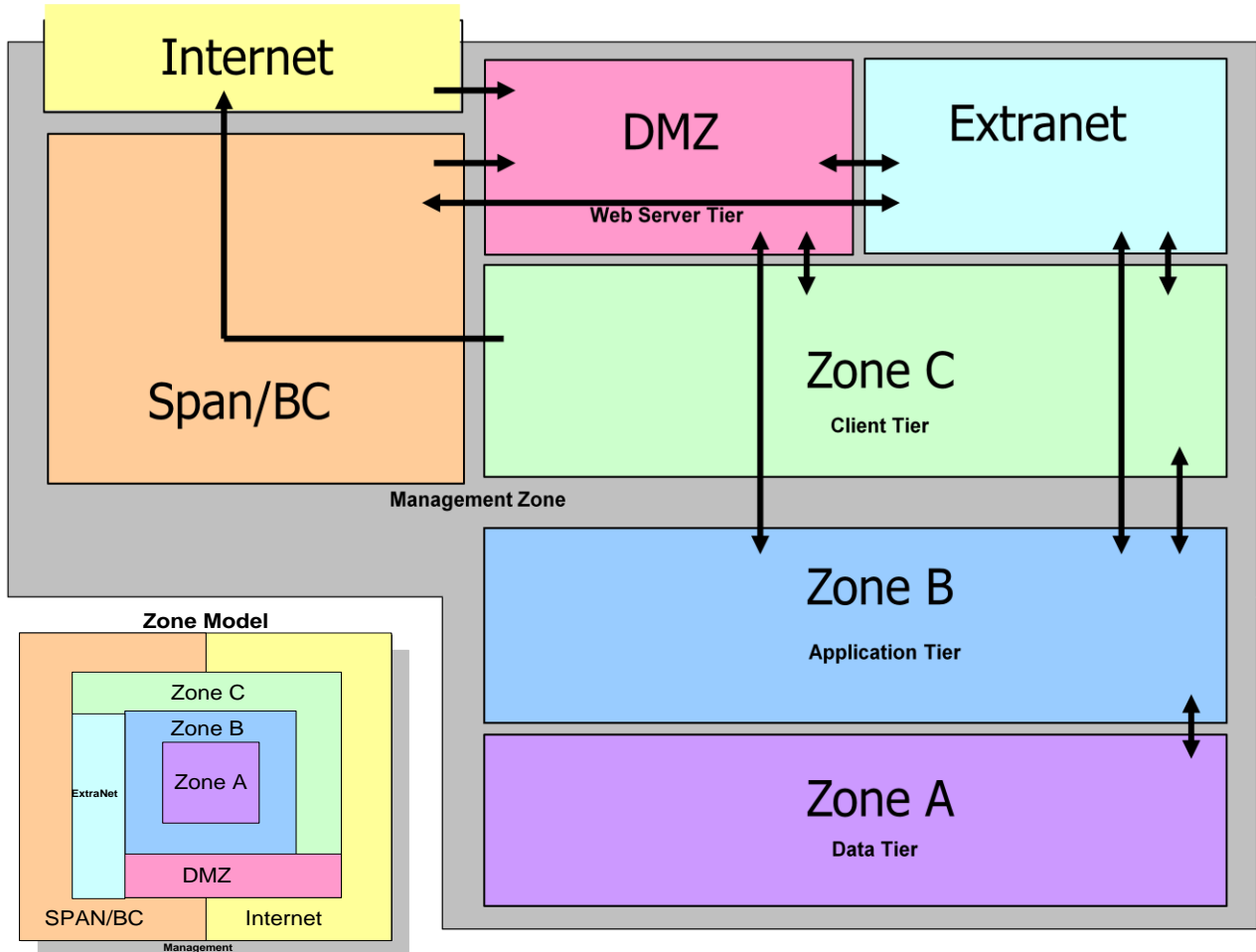
### ***A.4 Using an existing diagram as a starting point***

- Whole diagram
- Importing (package)
- Copying: copy deep
- Element copying – as link or as new



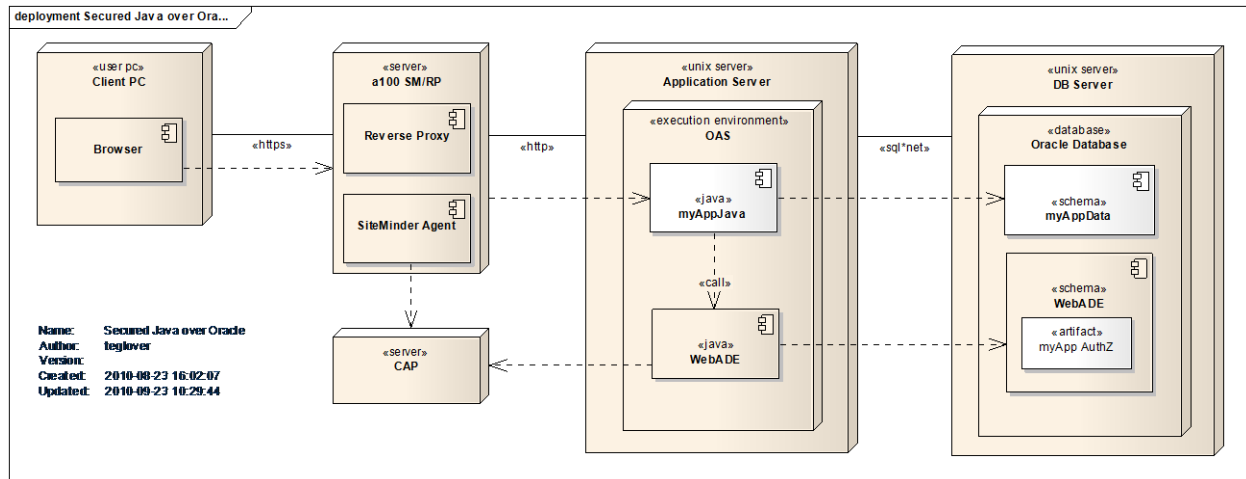
## Appendix B: Security Enhancement Project (SEP) Zonation

For reference, the SEP network zonation model (with its colour scheme) is reproduced here. It represents a target zonation toward which Ministries are working.

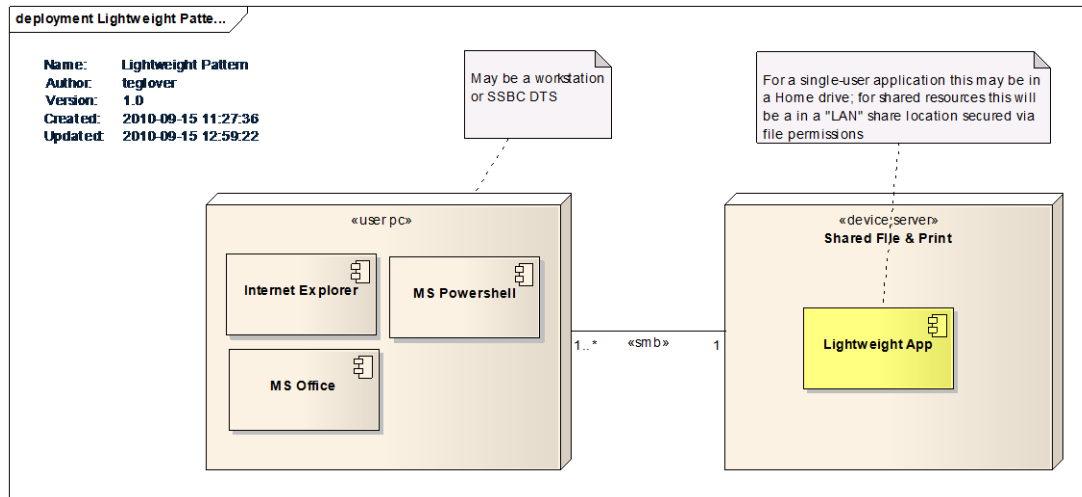


# Appendix C: Diagram Examples

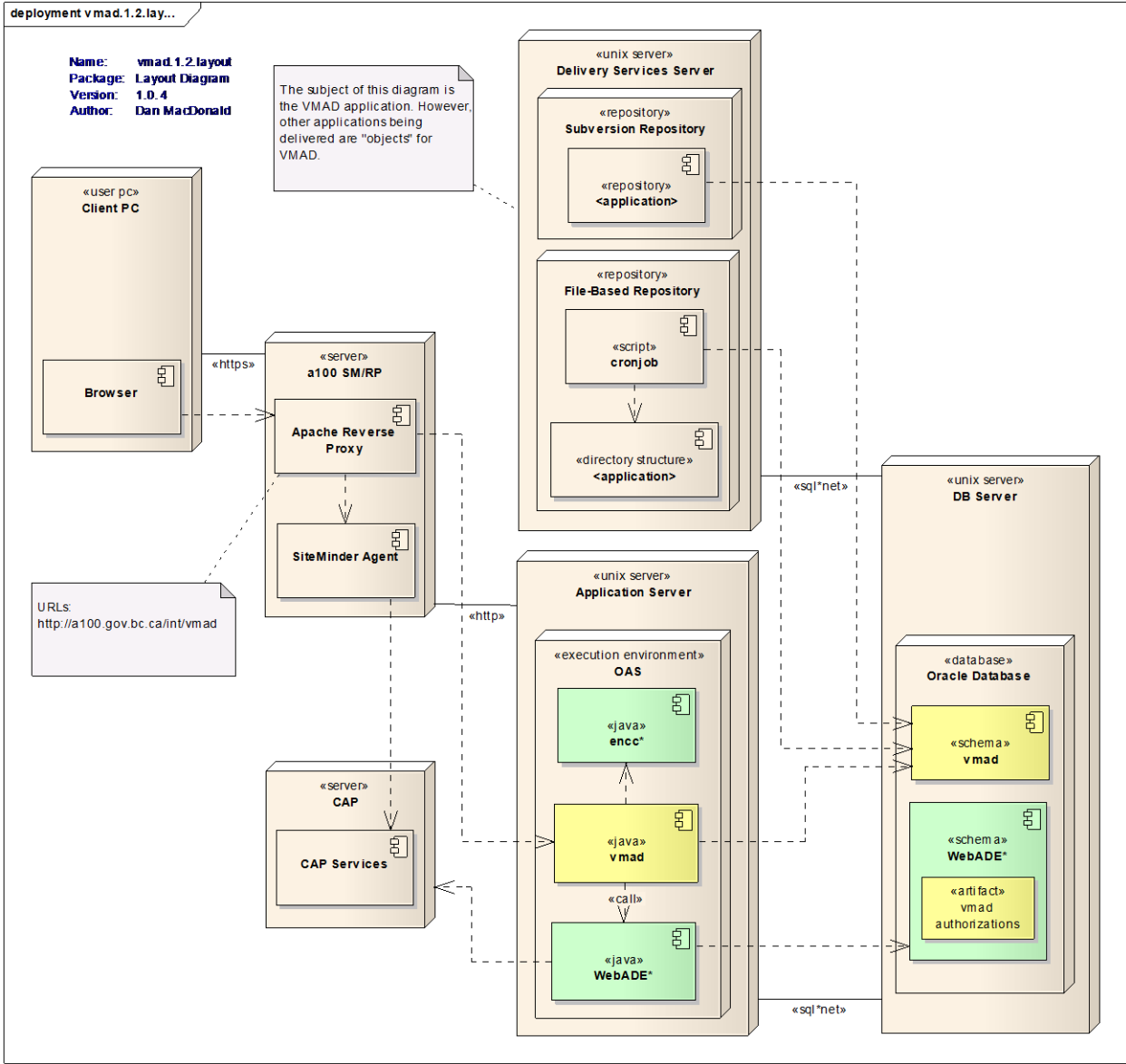
## C.1. Application Pattern: Secured Java over Oracle



## C.2. Application Pattern: Lightweight Development Approach



### C.3. Application Layout: VMAD





# C.5. Application Blueprint: SOFT / PROD

