

Arquitectura de Software

Deployment Viewpoint

Departamento de Ingeniería de Sistemas y Computación



Agenda del día



1. Deployment Viewpoint

2. Viewpoints / Views

3. Ejercicio

Deployment Viewpoint

- Usos

Describir el ambiente en el que el sistema será instalado y ejecutado, incluyendo las dependencias que el sistema tiene en el ambiente de ejecución

Deployment Viewpoit

- Concerns
 - Tipo de hardware requerido
 - Especificación y cantidad de HW requerido
 - Requerimientos de software de terceros (third-party)
 - Requerimientos de red (anchos de banda, etc.)
 - Compatibilidad con alguna tecnología
 - Restricciones físicas

Deployment Viewpoint

- Posibles Problemas durante su uso
 - Dependencias ambiguas
 - Tecnología desconocida / no probada
 - Falta de conocimiento tecnológico
 - Consideración del ambiente de despliegue tarde dentro del proceso

Deployment Viewpoint

- Stakeholders a los que está dirigido
 - Administradores del Sistema
 - Desarrolladores
 - Probadores (Testers)
 - Comunicadores
 - Asesores

Deployment Viewpoint - Aplicabilidad

- Normalmente ayuda a entender escenarios con:
 - Dependencias complejas en ejecución
 - Ambientes complejos de ejecución
 - Situaciones donde el sistema puede ser instalado en diferentes ambientes
 - Sistemas con requerimientos especiales y poco comunes de hardware o software

Deployment Viewpoint - Concerns

- Tipo de Hardware Requerido
 - Tipo de hardware y el rol que juega cada uno
 - Hardware de Ejecución
 - Hardware de Almacenamiento (soporte a BD)
 - Dispositivos de Red (Firewalls)
- Especificación y Cantidad del Hardware Requerido
 - Detalles específicos de HW
 - Modelo, Marcas, Serie

Deployment Viewpoint - Concerns

- Requerimientos de Software de Terceros
 - Software de terceros
 - Sistema Operacional
 - Plataforma de Ejecución
 - Librerías
 - Sistemas de Mensajería
 - Servidores de Aplicación
 - Servidores de Bases de Datos
 - Servidores Web
 - Otros...
 - Dependencias entre el sistema y los elementos listados en el punto anterior

Deployment Viewpoint - Concerns

- Compatibilidad Tecnológica
 - Requerimientos impuestos por cada uno de los elementos de hardware y software mencionados anteriormente
 - Ejemplo:
 - Rack de discos ópticos para almacenamiento de información (BD) que sólo funciona con el sistema operacional Windows Server (Enterprise Edition) SP2

Deployment Viewpoint - Concerns

- Requerimientos de Red / Capacidad
 - Decidir que elementos de hardware se necesitan y que elementos funcionales van a alojar
 - Tipo de tráfico esperado
 - Ancho de banda requerido
- Restricciones Físicas
 - Espacio para servidores / Rackolas de Discos
 - Temperatura
 - Distancias de Cableado

Deployment Viewpoint - Modelos

- Modelos utilizados para documentar este punto de vista
 - Modelos de plataforma de ejecución
 - Modelo de Red
 - Modelos de Dependencia de Tecnología

Deployment Viewpoint - Modelos

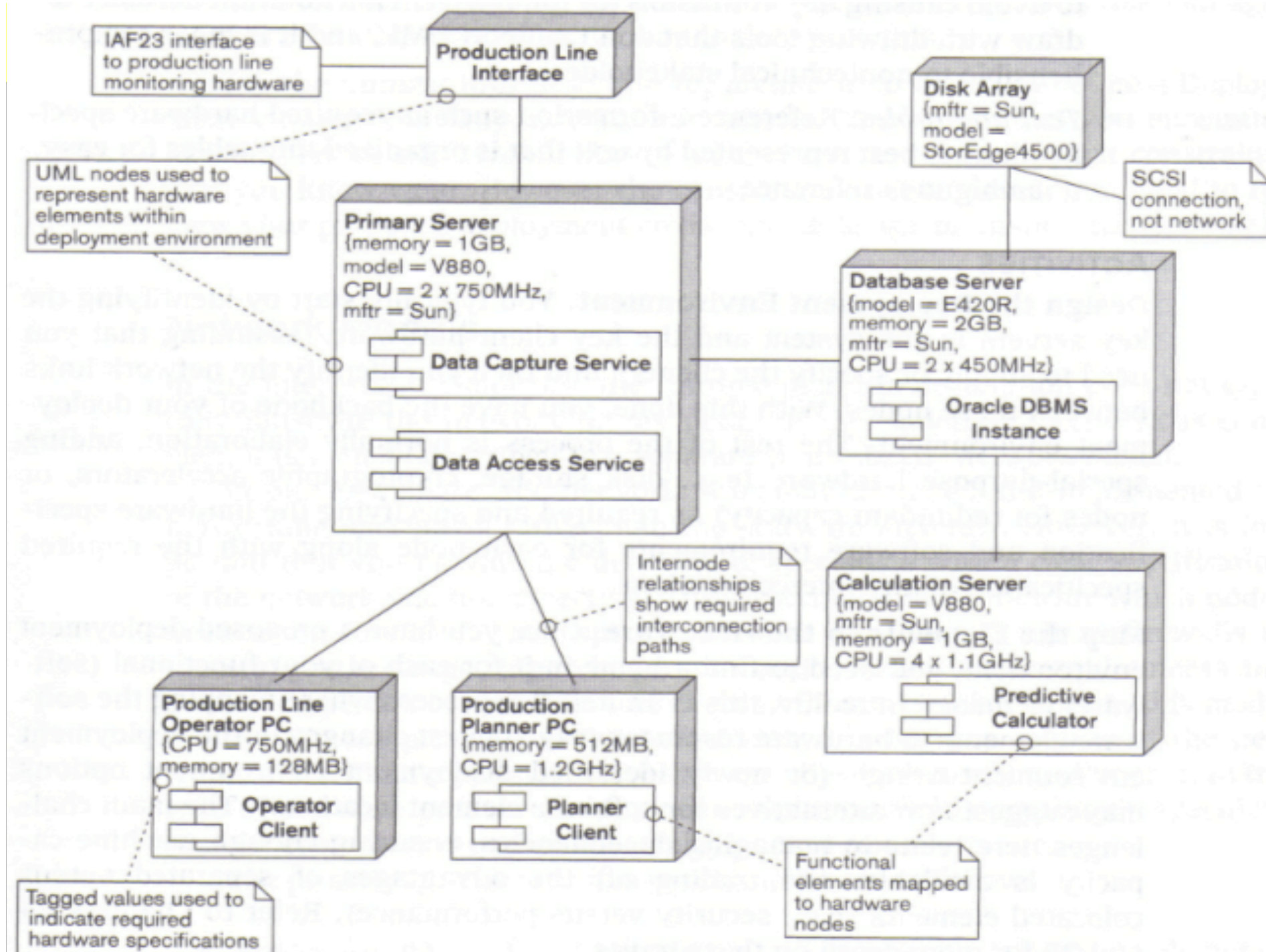
- Modelos de Plataformas de Ejecución
 - Descripción General
 - Modelo principal de este punto de vista
 - Define un conjunto de nodos (hardware)
 - Conecciones entre dichos nodos
 - Elementos de software que habitan cada uno de los nodos

Deployment Viewpoint - Modelos

- Principales Elementos
 - Nodos de procesamiento
 - Nodos Cliente
 - Hardware de almacenamiento en línea
 - Hardware de almacenamiento fuera de línea
 - Enlaces de Red
 - Otros componentes de Hardware
 - Relación entre elementos de ejecución y nodos

Deployment Viewpoint - Modelos

- Notación utilizada
 - UML - Diagrama de Despliegue (Deployment)
- Actividades a Desarrollar
 - Diseñar el ambiente de ejecución
 - Relacionar elementos de ejecución con nodos
 - Estimar los requerimientos de HW

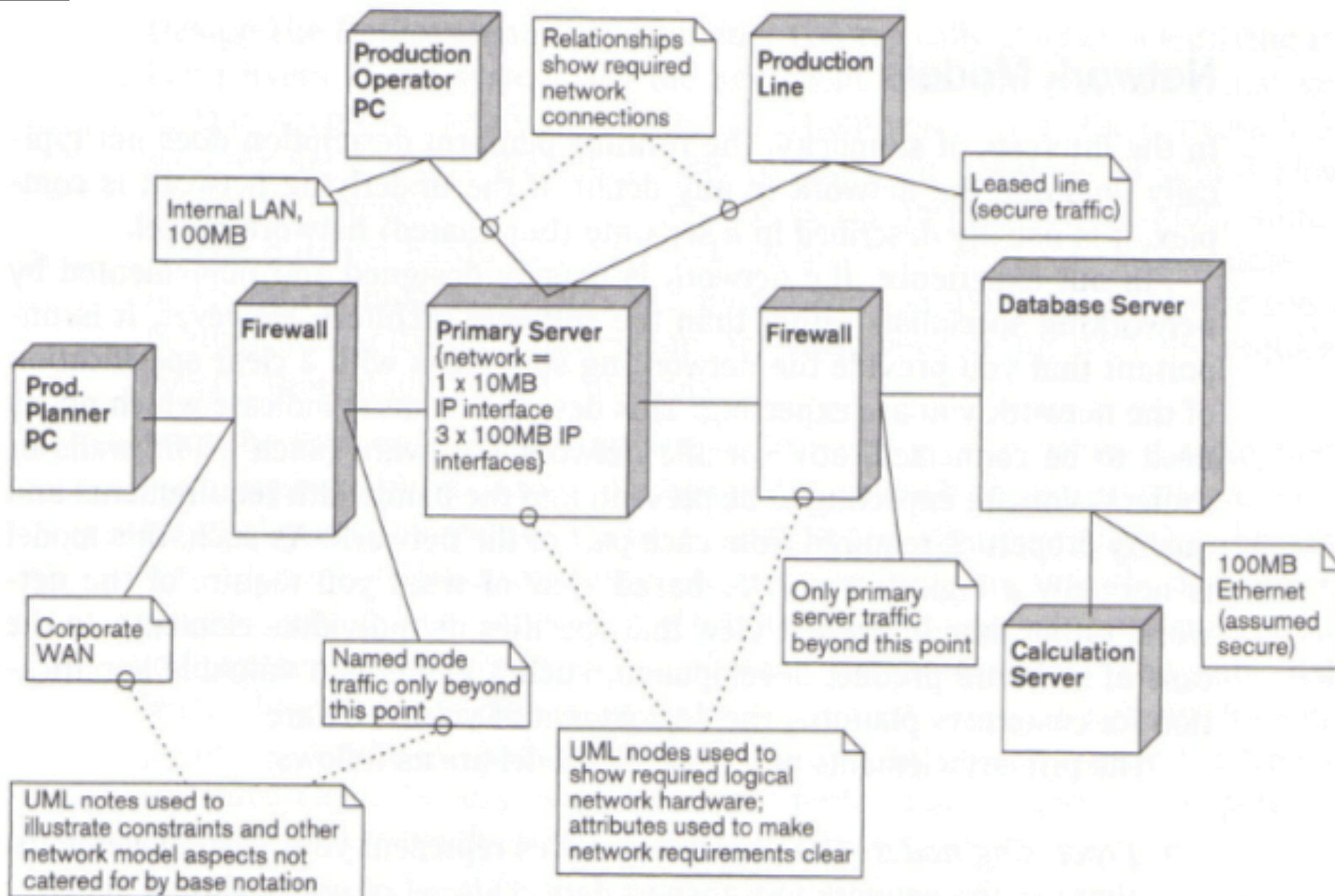


Deployment Viewpoint - Modelos

- Modelos de Red
 - Descripción General
 - Utilizado para describir en detalle la red asociada al modelo de plataformas de ejecución
 - Se debe especificar claramente
 - Los nodos que se conectan
 - Hardware de red específico (Firewall, Router, Bridge)
 - Requerimientos de ancho de banda
- Principales Elementos
 - Nodos de procesamiento
 - Nodos de red
 - Conexiones de Red

Deployment Viewpoint - Modelos

- Notación Utilizada
 - UML : Diagramas de Despliegue
- Actividades
 - Diseño de la Red
 - Estimación de la Capacidad



Deployment Viewpoint - Modelos

- Modelos de Dependencia de Tecnología
 - Descripción General
 - Definen las dependencias con el ambiente de ejecución (Sistema Operacional, Arquitectura de Procesadores, etc.)
 - Notación
 - Notaciones Gráficas
 - Tablas y Texto
 - Actividades
 - Analizar las dependencias de ejecución
 - Realizar una evaluación técnica

Deployment Viewpoint - Modelos

- Ejemplo

Componente	Requiere
Servicio de Acceso a Datos	Solaris 8.0.2 Sun C++ 4.1.2 Libraries
Servicio de Captura de Datos	Solaris 8.0.2 Sun C++ 4.1.2 Libraries Oracle OCI Libraries 8.1.7.3
Sun C++ 4.1.2	Solaris patch 1534567 Solaris pathc 1538367
Oracle OCI 8.1.7.3	Solaris optional module SUNWcipx Solaris patch 1583956
....	

Tomado de [1] pag 318

Caso de Estudio: Planning Tool

- dotProject es una herramienta web para planeación y seguimiento de proyectos (similar a MS project y otros)
- La interfaz web es bastante buena para ciertos stakeholders (p.e. los usuarios que quieren registrar tiempo)
- La interfaz NO es útil para los encargados de hacer la planeación (varios clicks + campos por cada tarea a crear * 100s de tareas = ineficiencia)

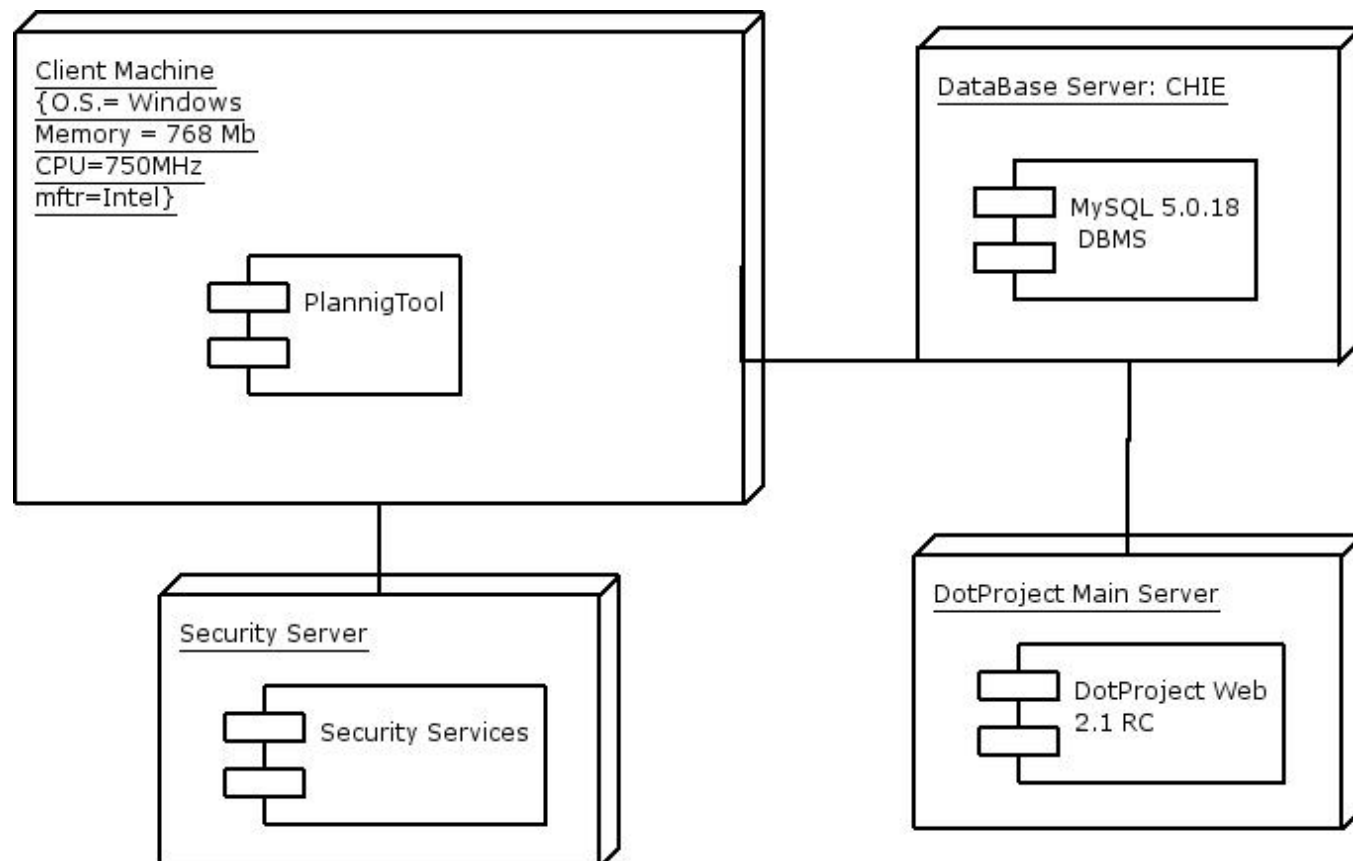
Caso de Estudio: Planning Tool

- PlanningTool es un módulo para dotProject
- Extiende la funcionalidad para permitir un mejor manejo de la planeación y seguimiento de proyectos de software
- Dentro de las funcionalidades que implementa están:
 - **Creación rápida de tareas por fase/tipo de proyecto, tipo de tarea, etc. (p.e. una reunión para todos los miembros solo es creada una vez y es replicada para todos los usuarios)**
 - **Manejo de reportes de seguimiento**
 - **Sincronización de datos con dotProject**

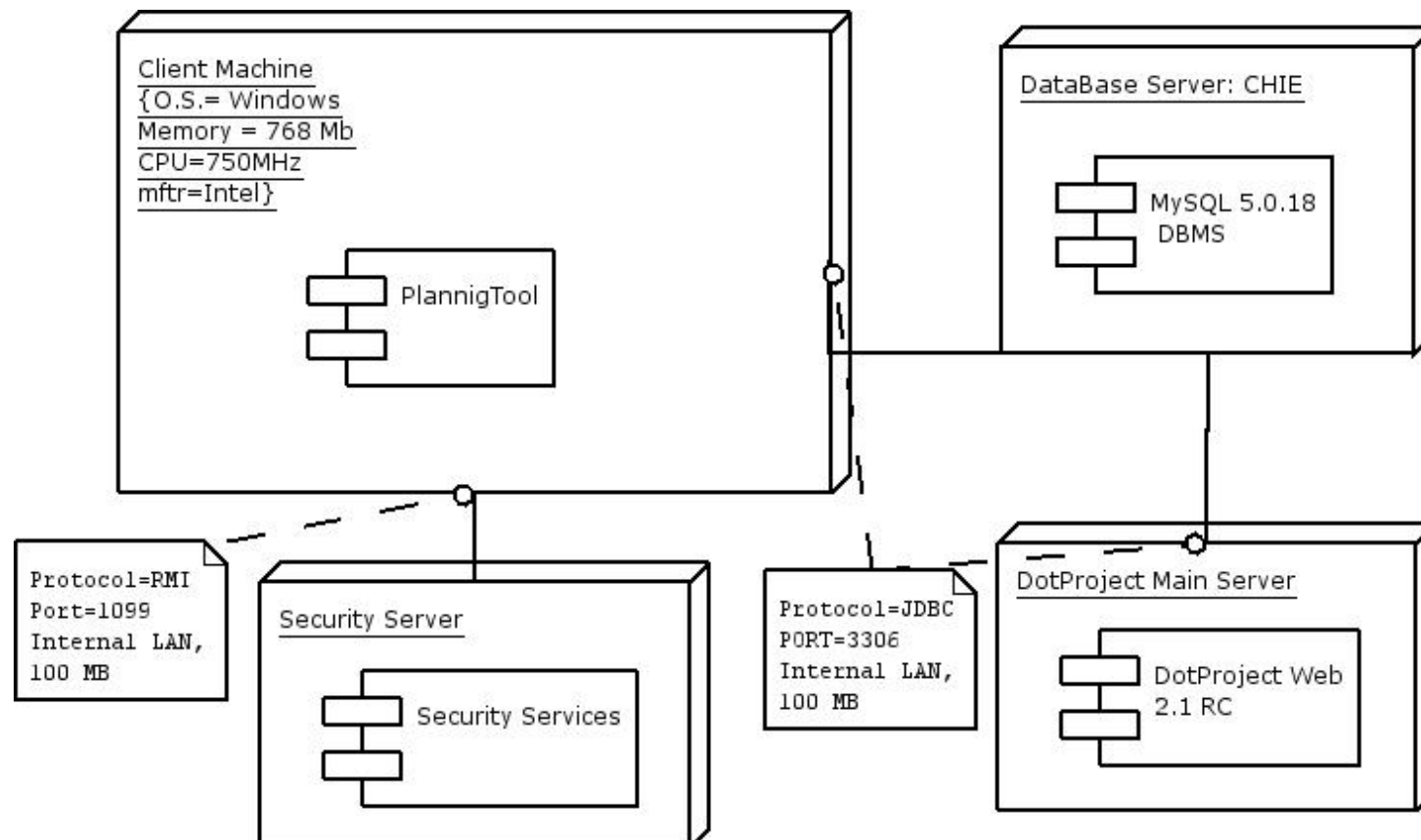
Caso de Estudio: Planning Tool

- Planning tool es una herramienta en Java simple. Sin embargo depende de muchas otras tecnologías:
 - DotProject es una herramienta PHP web con una base de datos MySQL
 - Las planeaciones son manejadas por la herramienta en archivos XML
 - La aplicación es altamente interactiva (MVC)
 - Los reportes deben poderse exportar a Excel
 - La seguridad del grupo Qualdev se realiza con un módulo de seguridad remoto (tecnología JEE)
 - Los reportes son fácilmente configurables gracias al uso de tecnología de Jasper Reports

Vista de Deployment (modelo de runtime simple)



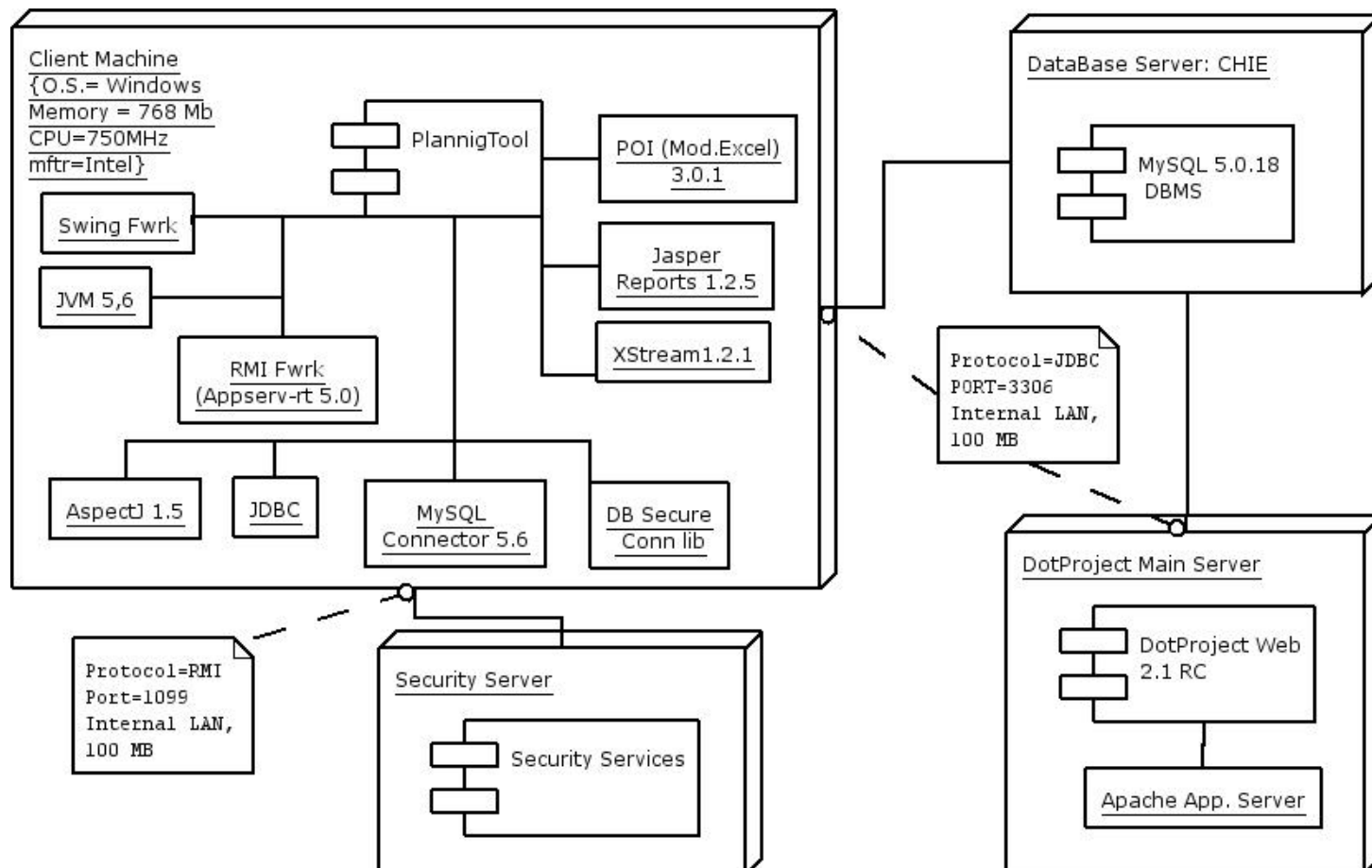
Vista de Deployment (modelo de runtime + red)



Preguntas

- Qué aspectos pueden verse en la segunda versión del diagrama que son ignoradas en la primera?
- Qué restricciones imponen?
- Es posible usar planning tool en un portátil con la red inalámbrica de los Andes?

Vista Deployment (completa)



Detalles de dependencias

- JVM 5, 6: La aplicación funciona correctamente con la versión 5 o 6 de la JVM
- Swing Framework: aplicación altamente interactiva
- RMI (Appserv-rt 5.0): conexión remota a módulo de seguridad
- AspectJ 1.5: Servicios de logging, captura de excepciones y monitoreo de avance de tareas
- JDBC: Framework de conexión genérica a base de datos en java
- MySQL Connector 5.6: Versión de JDBC y librerías para conexión a MySQL

Detalles de dependencias

- DB Secure Conn lib: Librería propietaria del grupo para establecer conexiones de manera segura a la base de datos del grupo
- POI (ModExcel) 3.0.1: Permite exportar reportes como archivos de excel
- Jasper Reports 1.2.5: Framework para definir y generar reportes
- XStream 1.2.1: Librería para manejo de archivos XML
- Apache App Server: Servidor web de dot project

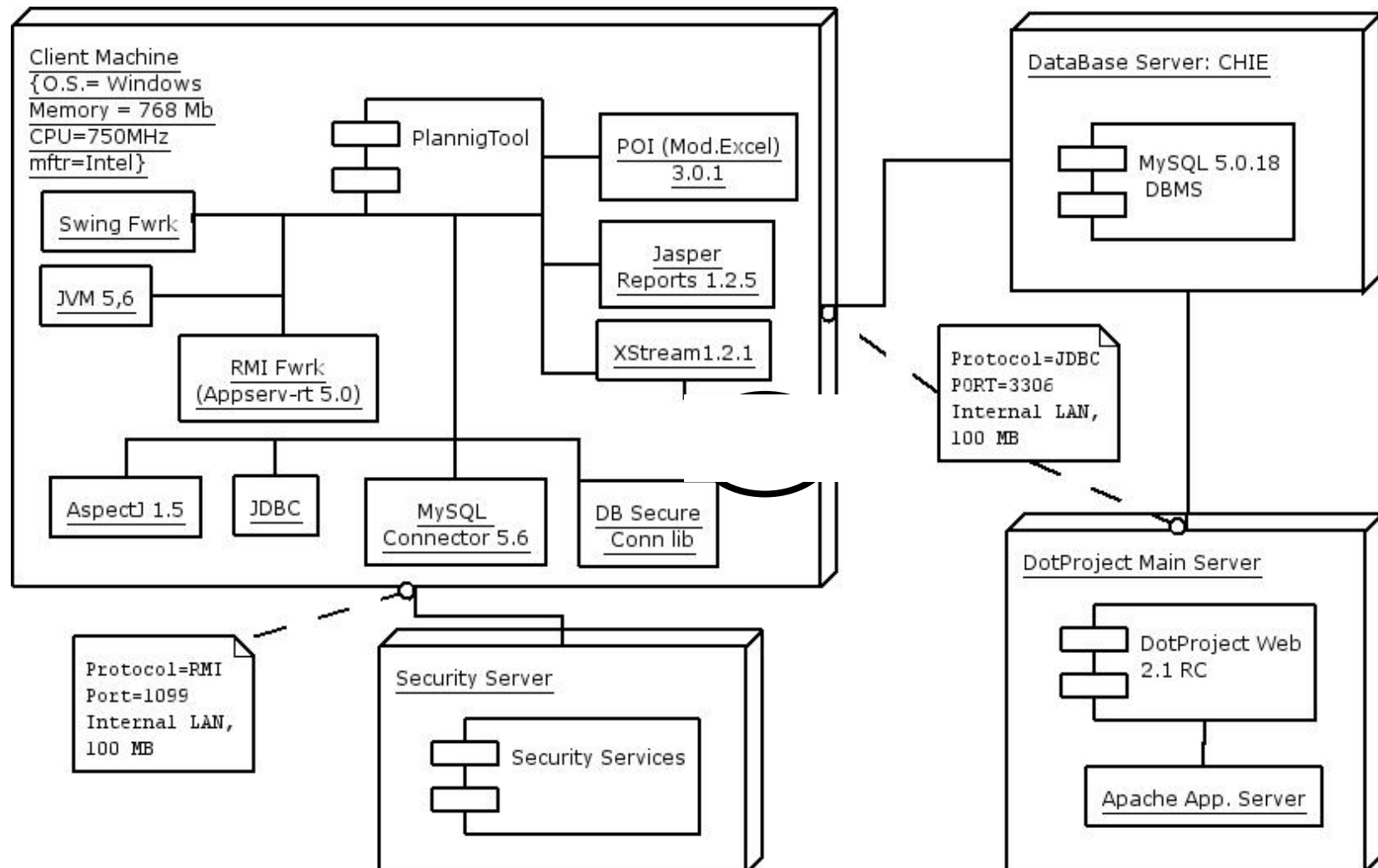
Detalles de dependencias

- Otras dependencias no son mostradas en el diagrama (para reducir complejidad):
 - Dependencias de Security Services (Glassfish, JEE, etc)
 - Otras dependencias de dotProject

Caso de Estudio: Planning Tool

- Planning tool en la versión presentada en el diagrama anterior tenía problemas arquitecturales:
 - Las planeaciones de 1000s de tareas causaban un error
- Volvamos al diagrama de deployment, alguna idea de porqué?

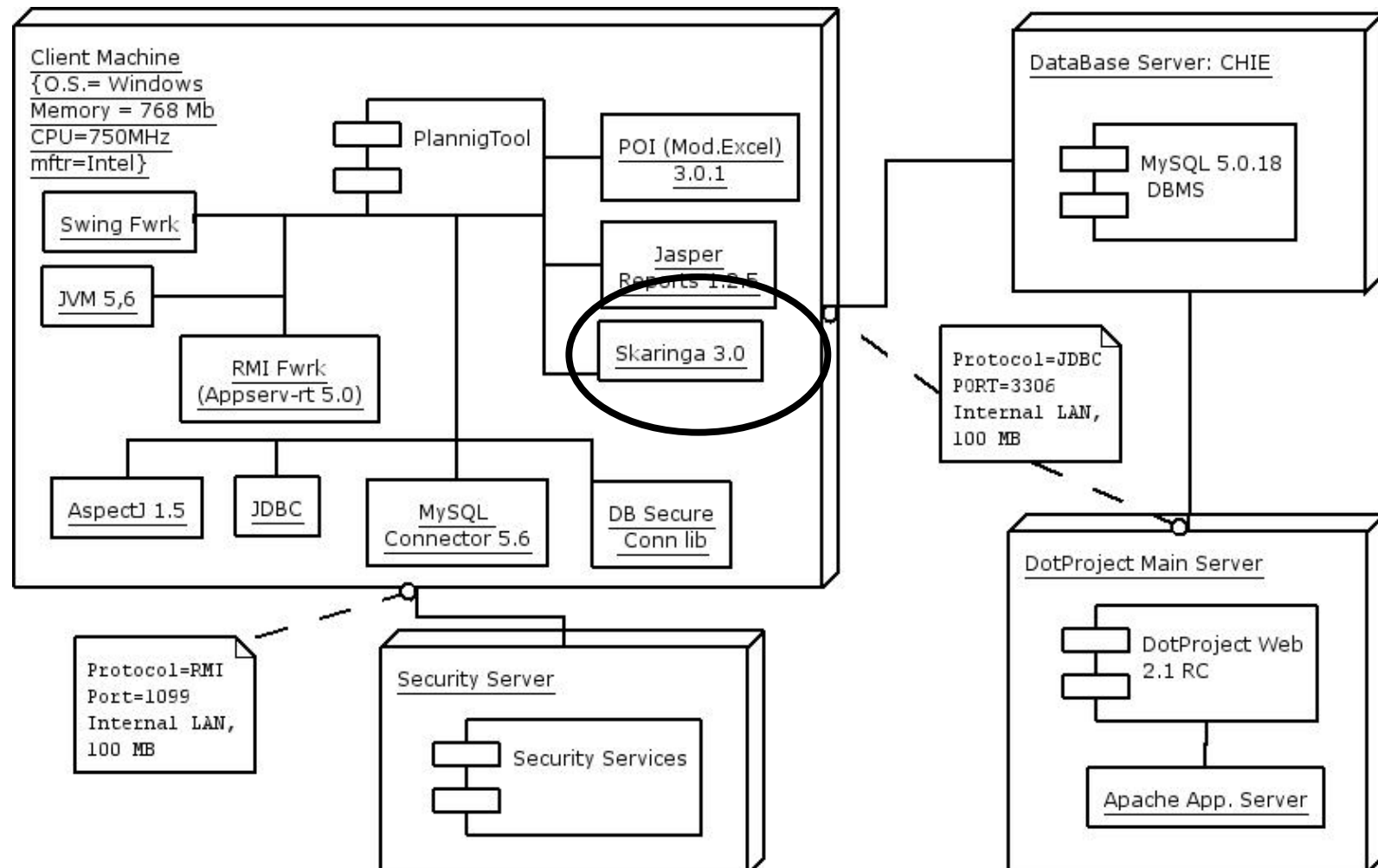
Vista Deployment (+ detalle)



Caso de Estudio: Planning Tool

- Xstream está basada en tecnología DOM
 - DOM no es escalable para archivos XML muy grandes porque carga en memoria TODO el archivo

Vista Deployment (nueva)



Caso de Estudio: Planning Tool

- SKaringa está basada en SAX
 - SAX carga el archivo solo a medida que lo lee

Caso de Estudio: Conclusiones

- Este diagrama permite validar si una configuración de mi sistema es adecuada
- Es útil como herramienta para entender una nueva tecnología ya que nos obliga a encontrar las dependencias que tiene (p.e. para JSF)
- El diagrama de deployment también ayuda identificar puntos en los que dependemos de tecnologías que pueden ser sensibles

Ejercicio (para preparar en casa)

- Elabore un diagrama de deployment para el laboratorio de JSF (debe ser muy similar al del taller 1)
- Los nodos y conexiones de su diagrama no deben ser muy complejos.
- Haga un primer diagrama con esta información que no tenga las dependencias de tecnología
- Revise qué tipo de conexiones se establecen entre los nodos, qué puertos usan, qué capacidad necesitarían (especule), etc.

Identificando dependencias de tecnologías

- Sin perder una copia del diagrama anterior ahora aumentelo con las dependencias de tecnología
- Revise las fuentes variadas de información que tiene del laboratorio
 - Qué versión del Jboss necesitamos usar, qué diferencia tendrá de las anteriores o posteriores
 - Qué contenedor de aplicaciones web está en el JBoss, qué versión en particular del contenedor está en el JBoss que usamos
 - Qué librerías que debe agregar al contenedor para el taller. Para qué sirven?
 - Qué librerías tiene el proyecto, cuales son estrictamente necesarias y para qué sirven
 - Revise la presentación de JSF y averigüe mas al respecto usando las referencias en la página del curso

Caracterizando dependencias tecnológicas

- Una vez identifique unas tres o cuatro librerías o frameworks de los que depende su proyecto investigue un poco más
 - Busque en google la librería
 - Para qué sirve, que servicios principales ofrece
 - Qué dependencias tiene esta librería para su correcto funcionamiento?

Para traer la próxima clase

- Traiga los dos diagramas que realizó :
 - El simple sin tecnología y el más complejo
- Traiga los resultados de la investigación que hizo sobre las dependencias
 - Resumen corto
 - Fuentes de información que usó
 - Foros a consultar en caso de que ocurran errores con esta tecnología
- Esté preparado para pasar al tablero con sus diagramas (Tráigalos en formato digital) y contarnos qué investigó

- Material preparado por
 - Darío Correal
 - Nicolás López

Bibliografía

- [1] Rozanski N, Woods E. “Software Systems Architecture” Addison-Wesley. 2005