

Arquitectura de Software

Vista de distribución

Índice

- Introducción
- Vista de despliegue
- Vista de instalación
- Vista de asignación de trabajo

Introducción

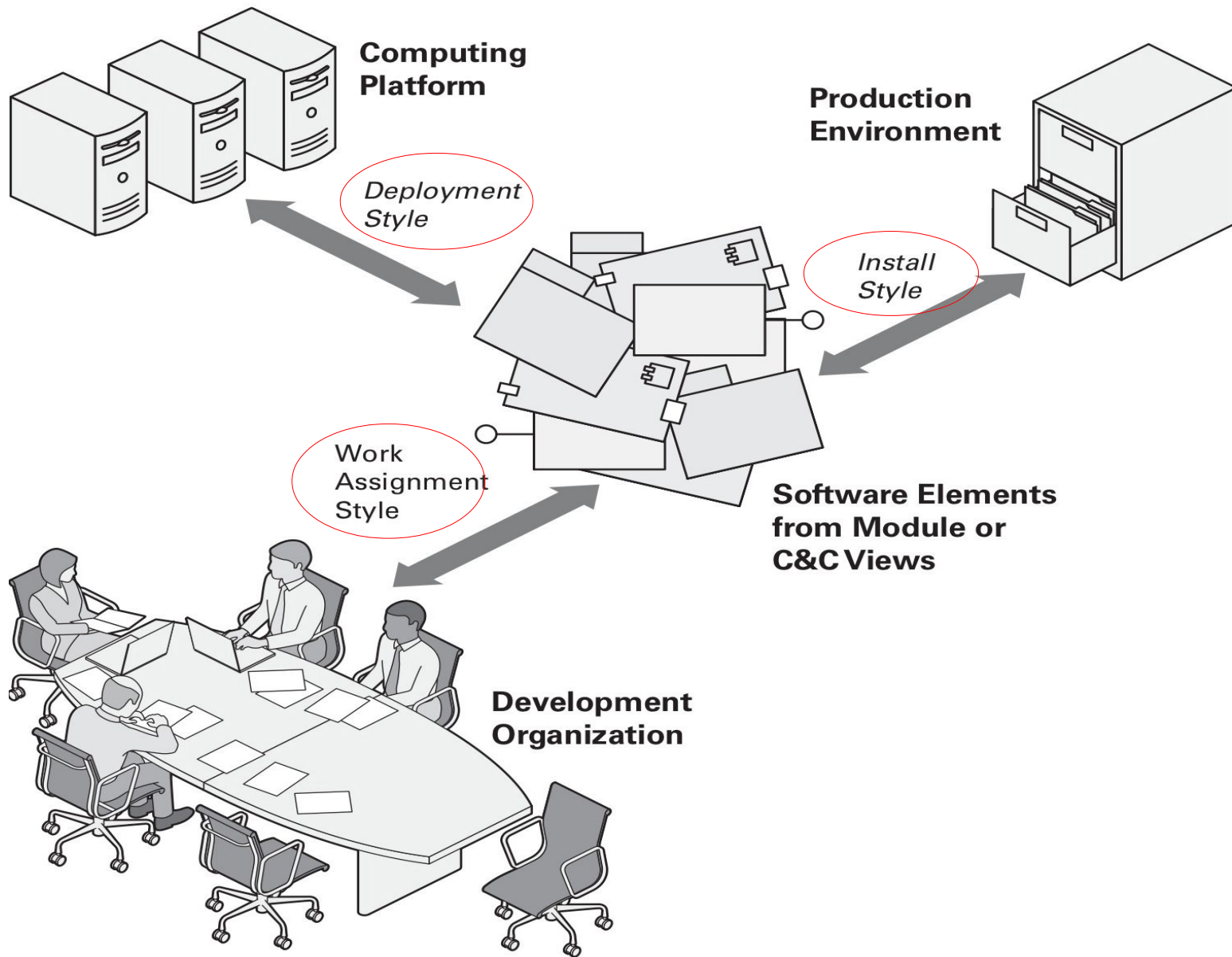
- La vista de distribución establece las relaciones entre la arquitectura del software y los elementos no software, el entorno
- El **entorno** es donde el software se desarrolla, despliega y ejecuta
- Elementos del entorno:
 - Hardware y redes
 - Sistemas de gestión de ficheros
 - Equipos de desarrollo

Introducción ...

- ¿Por qué?
 - Arquitectura software q Hardware
 - Analizar las prestaciones del sistema
 - Arquitectura software q Estructura de ficheros
 - Gestión del sistema en producción
 - Arquitectura software q Equipos de trabajo
 - Gestión de las actividades del proyecto

Introducción ...

- Esas relaciones son de “distribución” o de “asignación”
- Distinguimos tres Vistas
 - Despliegue
 - C&C q Hardware donde se ejecutará el software
 - Instalación
 - Componentes q Estructuras del sistema de ficheros en el entorno de producción
 - Asignación de trabajo
 - Módulos q Gente, equipos, unidades de



Introducción ...

Consideraciones

- Un elemento de la arquitectura se puede asignar a múltiples elementos del entorno y viceversa
- Si la **distribución** cambia a lo largo del tiempo (en desarrollo o en ejecución) la arquitectura es **dinámica** con respecto a esa distribución (Despliegue dinámico, ...)

Introducción ...

- Consideraciones
 - El objetivo de la vista de distribución es comparar las propiedades que requiere el elemento software con las propiedades que proporciona el elemento del entorno q ver si las satisface

Vista de despliegue

- Los elementos de la vista de C&C se asignan al hardware de la plataforma de computación sobre la que se ejecutarán

despliegue ...

- Elementos software
 - Asumimos que son entidades de ejecución: procesos, threads, puertos, memoria compartida
- Elementos del entorno
 - Hardware: procesadores, memoria, discos, redes, firewalls, routers, etc.
- Una *asignación válida* asegura que los *requisitos* (*propiedades*) expresados por los elementos software son satisfechos por las características de los elementos hardware

despliegue ...

- **Propiedades de los *elementos software***
 - Consumo de recursos
 - P.e. sabemos que una computación ejecuta 32.123 instrucciones (siempre, como máximo, de media, ...)
 - Requisitos de recursos y restricciones a satisfacer
 - Un elemento sw se debe ejecutar en menos de 0.1 seg
 - Condiciones de "disponibilidad"
 - Un elemento sw debe estar ejecutándose continuamente
 - Qué debe ocurrir para que migre el elemento software: lo vemos más adelante

despliegue ...

- **Propiedades de los *elementos del entorno***
 - Propiedades de la **CPU**
 - Velocidad del procesador, número de cores, tamaño de la caché, ...
 - Propiedades de la **memoria**
 - Tamaño y velocidad
 - Propiedades de los **discos**
 - **Ancho de banda**
 - **Tolerancia a fallos**

despliegue ...

- Relaciones

- La ya dicha de "asignación". En qué unidades físicas residen los elementos software en un momento determinado
- Si es **dinámica**, el **componente se puede mover de un procesador a otro**: *moverse de una maquina a otra*
 - **Migrar**: pero no está simultáneamente en ambos. Ejemplo: agentes móviles
 - **Copiar-migrar**: se envía una copia al nuevo procesador y se retiene la actual. Ejemplo: applets
 - **Ejecutar-migrar**: la copia está en varios procesadores pero sólo uno está activo en un momento determinado. Ejemplo: para alta disponibilidad

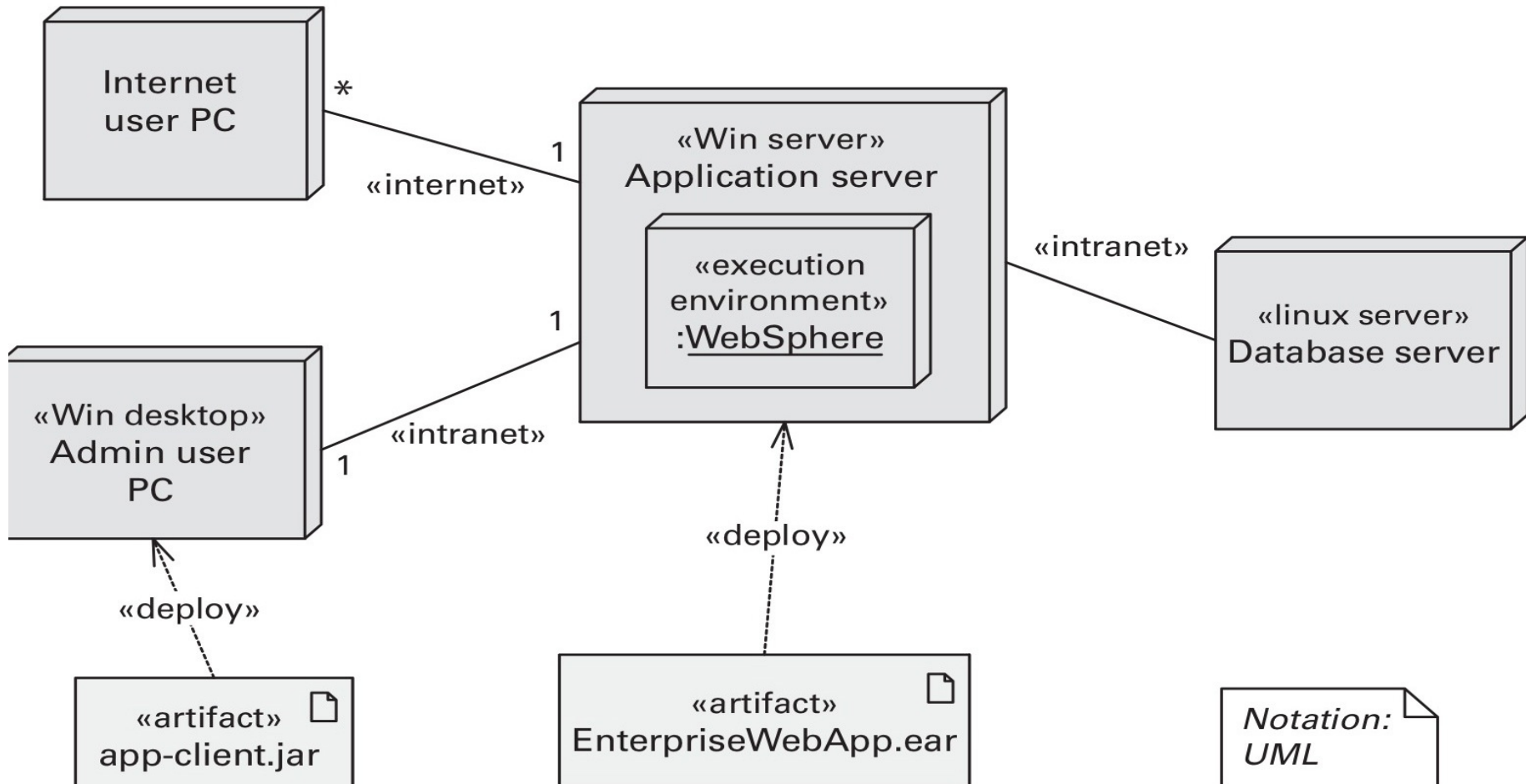
despliegue ...

- ¿Para qué sirve?
 - Analizar las prestaciones, disponibilidad, la fiabilidad y la seguridad del sistema
 - Para entender las dependencias en tiempo de ejecución q Pruebas del sistema
 - Para realizar la integración del sistema y confeccionar las pruebas de integración
 - Para estimar el coste del hardware

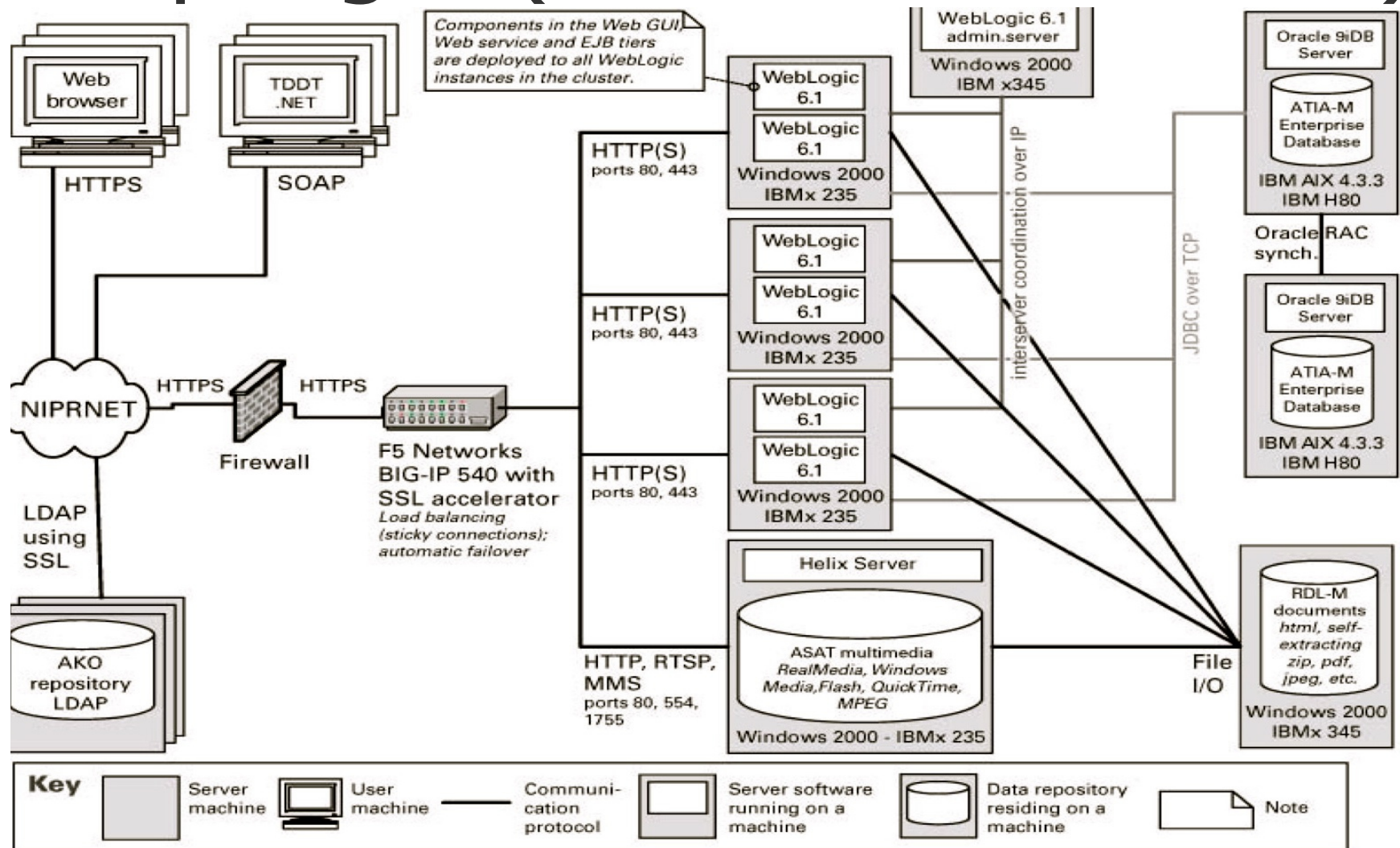
despliegue ...

- Notación UML
 - Diagrama de despliegue
 - Nodos (representados por cubos) y Componentes
 - Nodos:
 - Elementos de procesamiento Hw
 - Atributos: processorSpeed=1GHz, memory=128MB
 - Conectados por asociaciones que representan comunicaciones. Se pueden estereotipar
 - Tienen componentes que se ejecutan en ellos
 - Componentes:
 - Tienen dependencias entre ellos

despliegue ...



despliegue(notación informal)



Puntos a discutir

1. Compara la vista de despliegue con un “diagrama de red” creado por el administrador de la red
2. Supón que tu sistema va a ser desplegado en diferentes plataformas y configuraciones ¿Cómo lo representarías?

Vista de instalación

mapear componentes en sist ficheros

- En grandes sistemas Sw el número de ficheros instalados en el entorno de producción puede ser de cientos Ñ
 - Ejemplos: ejecutables, librerías, ficheros de datos, logs, ficheros de configuración, ficheros de control de versiones, ficheros de ayuda, ficheros con las licencias, HTML, scripts, imágenes, etc. etc.
- Además podemos tener versiones para diferentes plataformas

instalación ...

- Debemos agruparlos/empaquetarlos para instalarlos en el entorno de producción
 - ¿Cómo?: Técnicas de gestión de configuraciones, Herramientas de instalación (Asignatura Proyecto Software)
- La vista de instalación:
 - Asigna elementos (componentes, items de configuración, ...) al *sistema de gestión de ficheros* en el entorno de producción
 - Muestra cómo el sistema ya instalado está organizado en ficheros y directorios

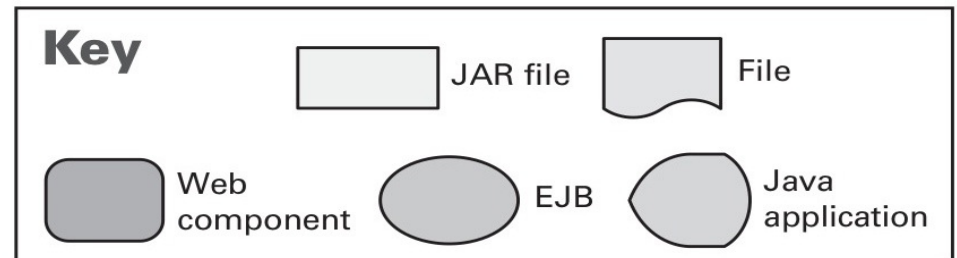
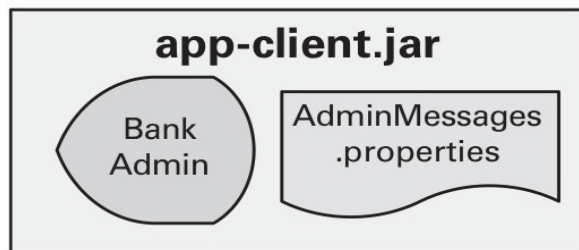
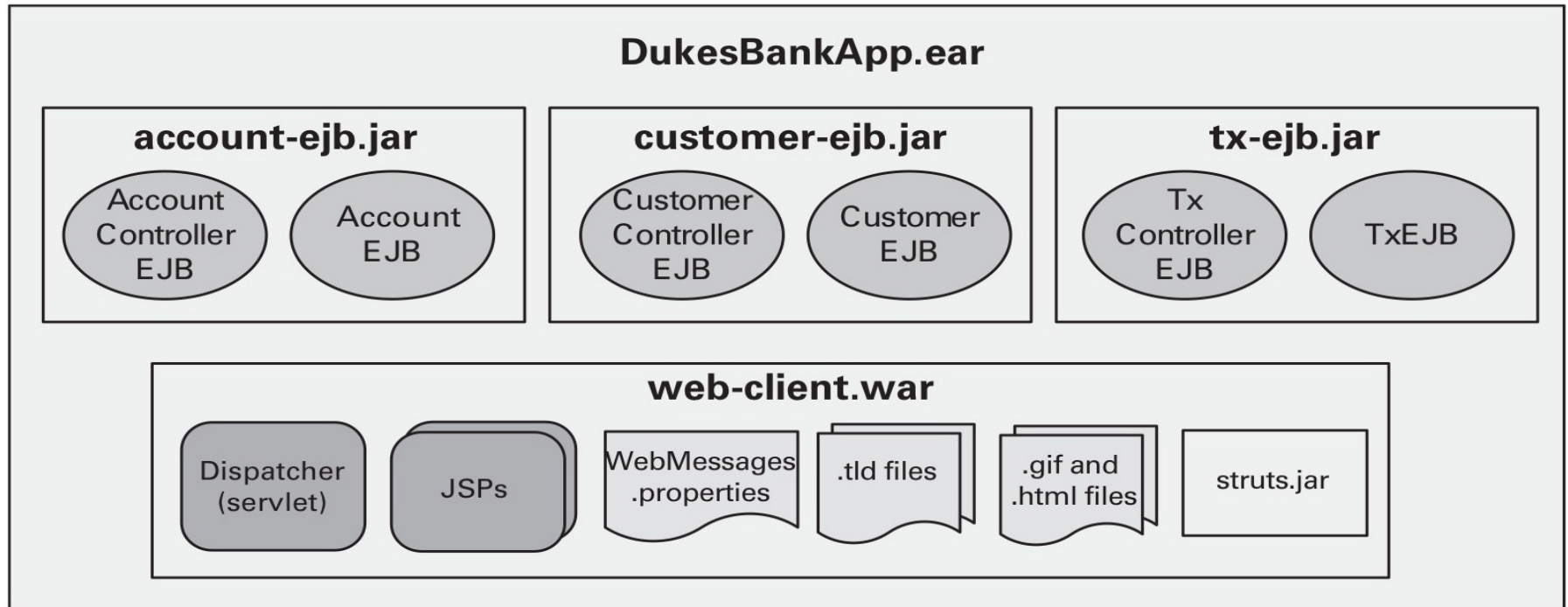
instalación ...

- Puede ser necesario crear diferentes versiones del sistema:
 - Diferentes idiomas
 - Diferentes precios (versión gratuita, comercial, pro)
 - Clientes con versiones específicas
- Esta vista ayuda a describir qué ficheros hay que usar y cómo se deben *configurar* para producir esas versiones

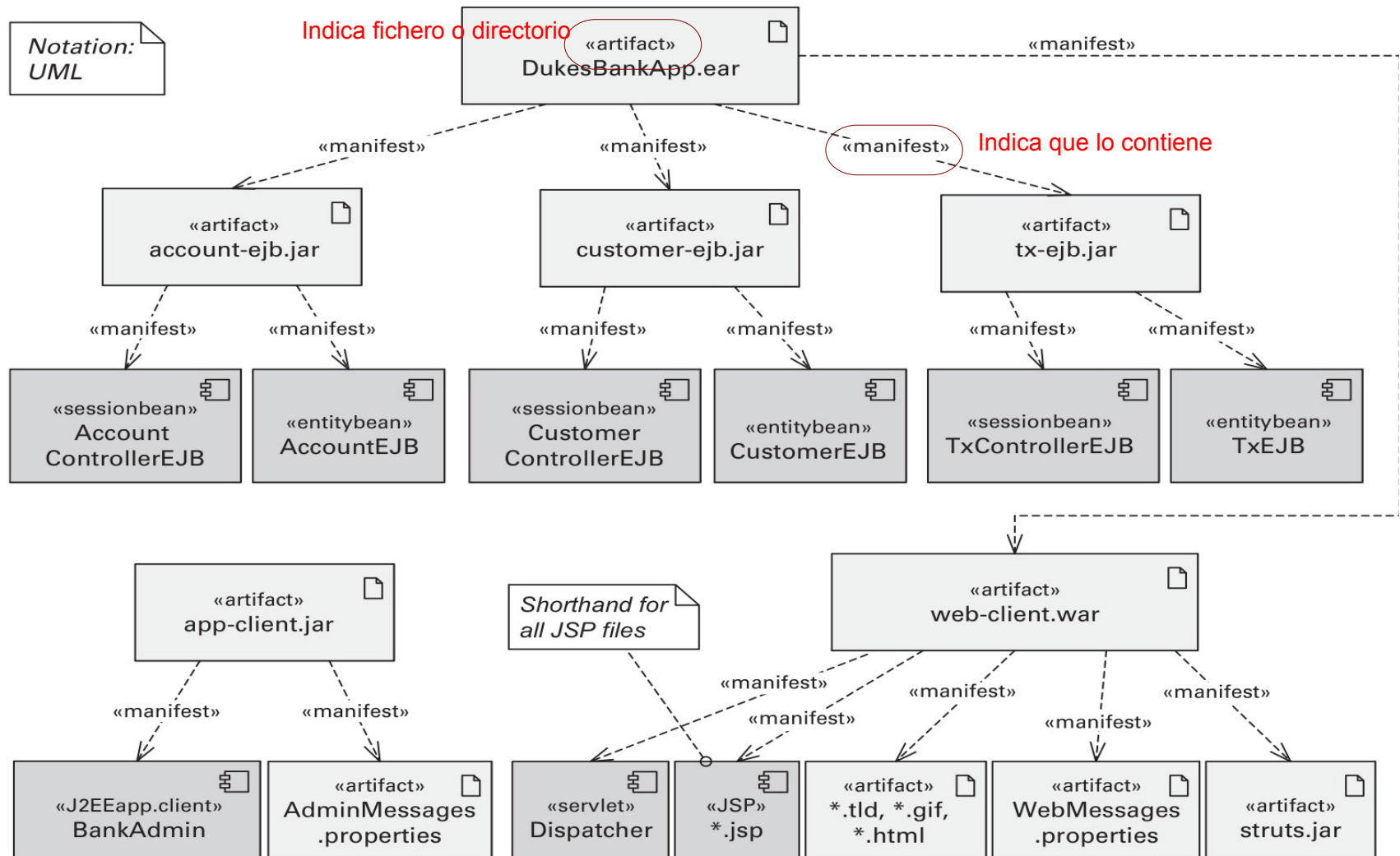
instalación ...

- ¿Para qué sirve?
 - Debe ayudar a los desarrolladores, instaladores y operadores del sistema a:
 - Crear los *procedimientos* para construir el sistema (*build* tool) y para desplegarlo
 - Mantener el control y la integridad del proceso de empaquetado y sobre el sistema desplegado
 - Navegar a través de los cientos de ficheros instalados (localizar logs, ficheros de instalación, ficheros dañados ...)
 - Seleccionar y configurar ficheros para crear versiones específicas
 - Actualizar y configurar ficheros de las diferentes versiones
 - Diseñar e implementar actualizaciones automáticas
- Hay herramientas para gestión de configuraciones, para hacer el *build* y para instalar

Ejemplo (notación informal)



Ejemplo (UML)



Vista de asignación de trabajo

- Asigna *módulos* a equipos o personas que serán responsables de llevarlos a cabo
entrada: vista de modulos
- Unimos actividades a recursos
- Pregunta: ¿Cuál es la entrada para esta vista?

Asignación de trabajo ...

- ¿Qué tareas/actividades asignamos?
 - Implementar e integrar los módulos
 - Pruebas de los módulos
 - Gestión de configuraciones
 - Código para construir y mantener el sistema
 - Implementación de la BBDD
 - ...

Asignación de trabajo ...

- Pregunta

- Si compramos un módulo en el mercado
componentes of the set (COTS) no habrá ninguna tarea de
implementación
 - ¿Nos debemos olvidar de este módulo
en esta vista?

no, si aparece, hay que integrarlo, probarlo...

Asignación de trabajo ...

- Relacionamos módulos con unidades de trabajo
 - Propiedades de los *elementos software*: requieren "habilidades" para llevar a cabo el trabajo
 - Propiedades de los *elementos del entorno*: poseen "habilidades"
- Muestra las grandes unidades de trabajo y quien las realizará, pero también las herramientas que se utilizarán

Asignación de trabajo ...

ECS Element (Module)		Organizational Unit
Segment	Subsystem	
Science Data Processing Segment (SDPS)	Client	Science team
	Interoperability	Prime contractor team 1
	Ingest	Prime contractor team 2
	Data Management	Data team
	Data Processing	Data team
	Data Server	Data team
	Planning	Orbital vehicle team
Flight Operations Segment (FOS)	Planning and Scheduling	Orbital vehicle team
	Data Management	Database team
	User Interface	User interface team
...

Asignación de trabajo ...

- Estructura de descomposición del trabajo (EDT)
 - Presentación simple y organizada del trabajo a realizar en el proyecto
 - ¿Cómo?
 - Descomposición jerárquica
 - Cada nivel descendente representa un detalle incrementado del trabajo del proyecto
 - Cada nivel es el resultado de la integración del siguiente
 - Orientada al *entregable*
 - Resultado del trabajo, no el trabajo en sí
 - Existen muchas formas/criterios de descomposición jerárquica
 - Si es por módulos la vista de módulos sería la EDT (los módulos son entregables)
 - EDT documenta el alcance del proyecto no su ejecución

Asignación de trabajo ...

- Discusión: ¿Este estilo es responsabilidad del arquitecto o de los gestores del proyecto?
 - Esta vista debe *ayudar y ser utilizada* por el gestor del proyecto
 - Planificar los recursos/personas
 - Asignar responsabilidades
 - Para EDT, planificación, presupuestos, nuevas contrataciones, ...

Relación con otras vistas

- Con la vista de módulos
 - La extiende añadiendo módulos que corresponden con las herramientas de desarrollo, pruebas, gestión de configuraciones q asignándolas
 - Opción económica: aumentar la vista de módulos con una nueva propiedad de los módulos q quién lo tiene asignado

