

Integrated Tools for On-Line Education

Professor Chris Jesshope
Massey University

Goals of the Tile project

- Tile is a project funded by the New Zealand's New Economy Research Fund
- ✔ The goals of the project are to:
 - design a flexibly delivery vehicle for on-line and off-line learning
 - to monitor and advise students effectively in whatever mode they are browsing
 - integrate a variety of authoring tools for ease of use
 - provide a configurable system that can be adapted to different learning environments

Flexible delivery

✓ The following modes of studying will be supported in Tile:

✓ *On-line with local server application*

- material mostly from CD or DVD
- the student still has access to the education provider's on-line archive
- local information can be updated from this source.

✓ *Off-line with local server application*

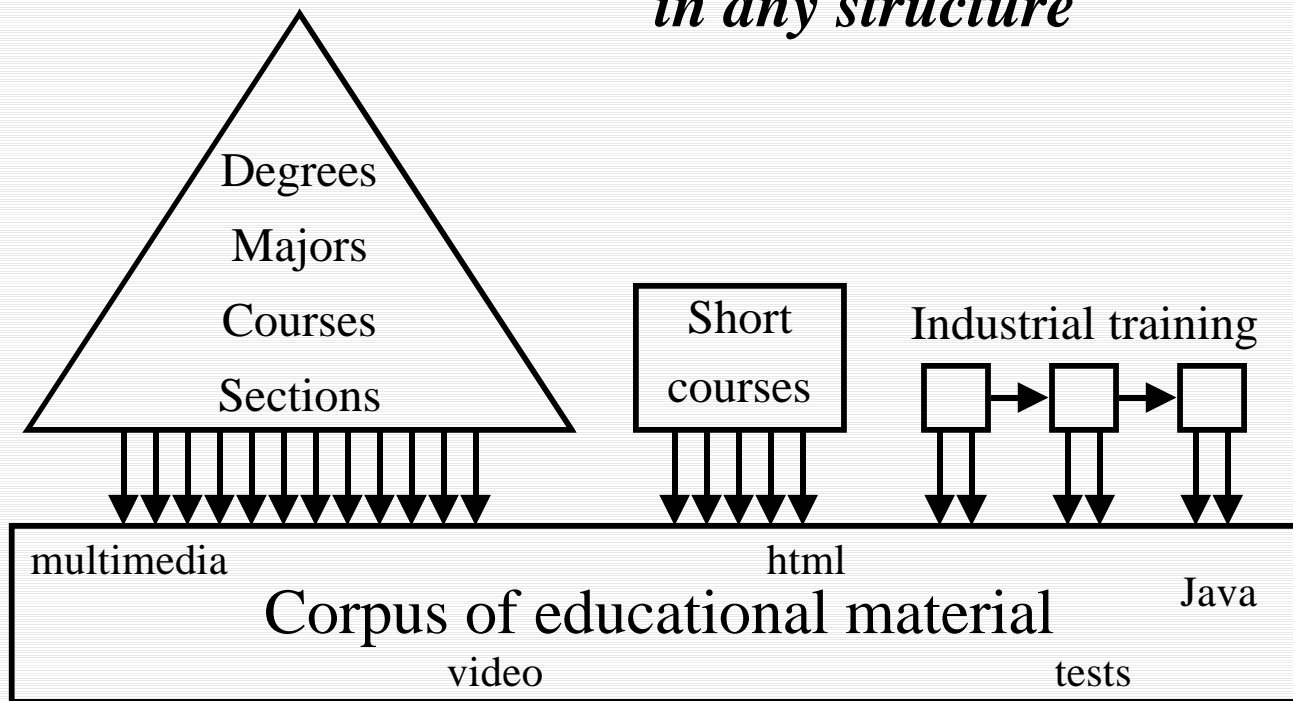
- no access to on-line material
- a local server continues to monitor the student's progress

✓ *On-line with no local server application*

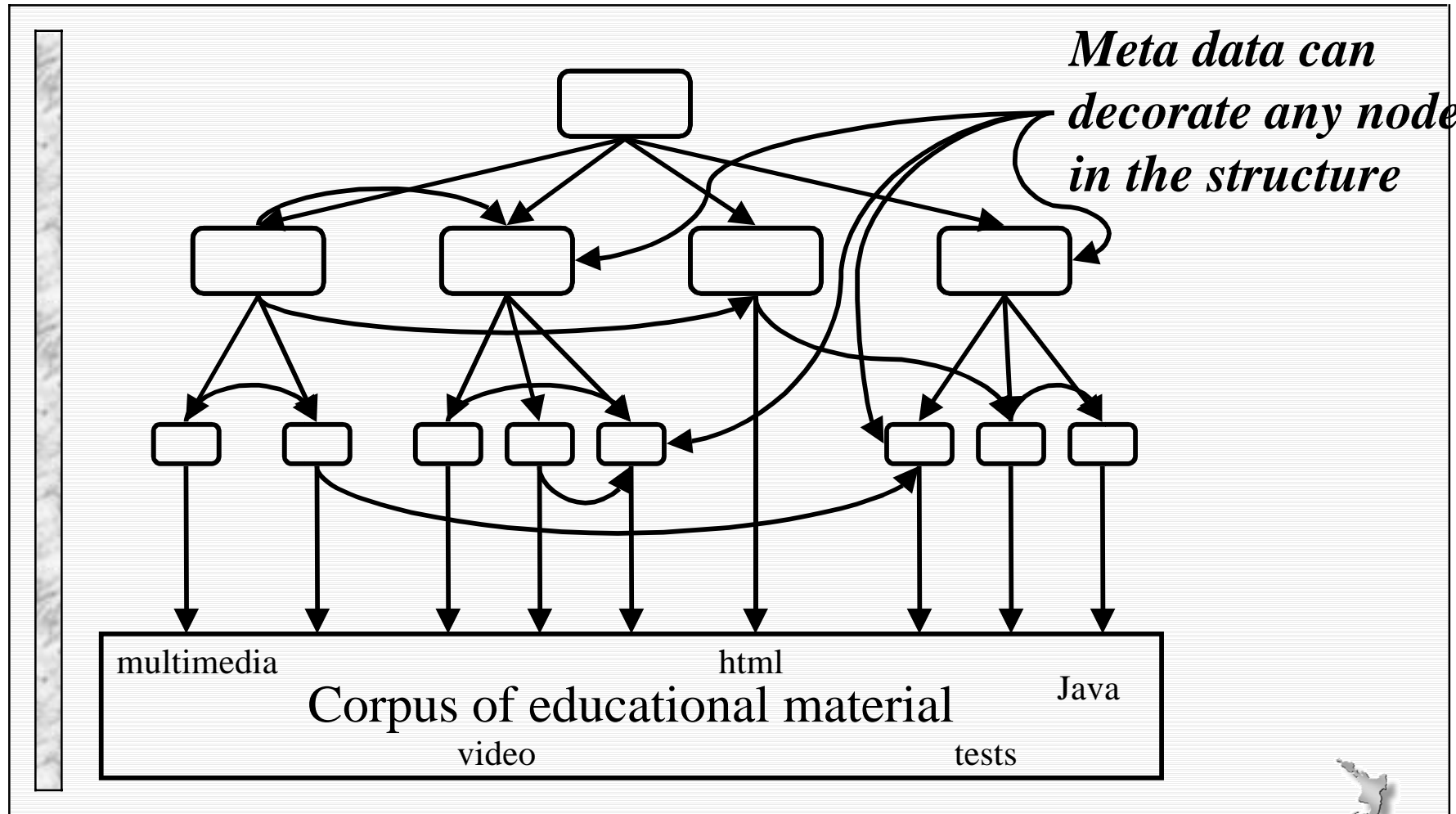
- no local software
- access and monitoring by the education provider's remote server

Flexible structure

The same educational material may be reused in any structure



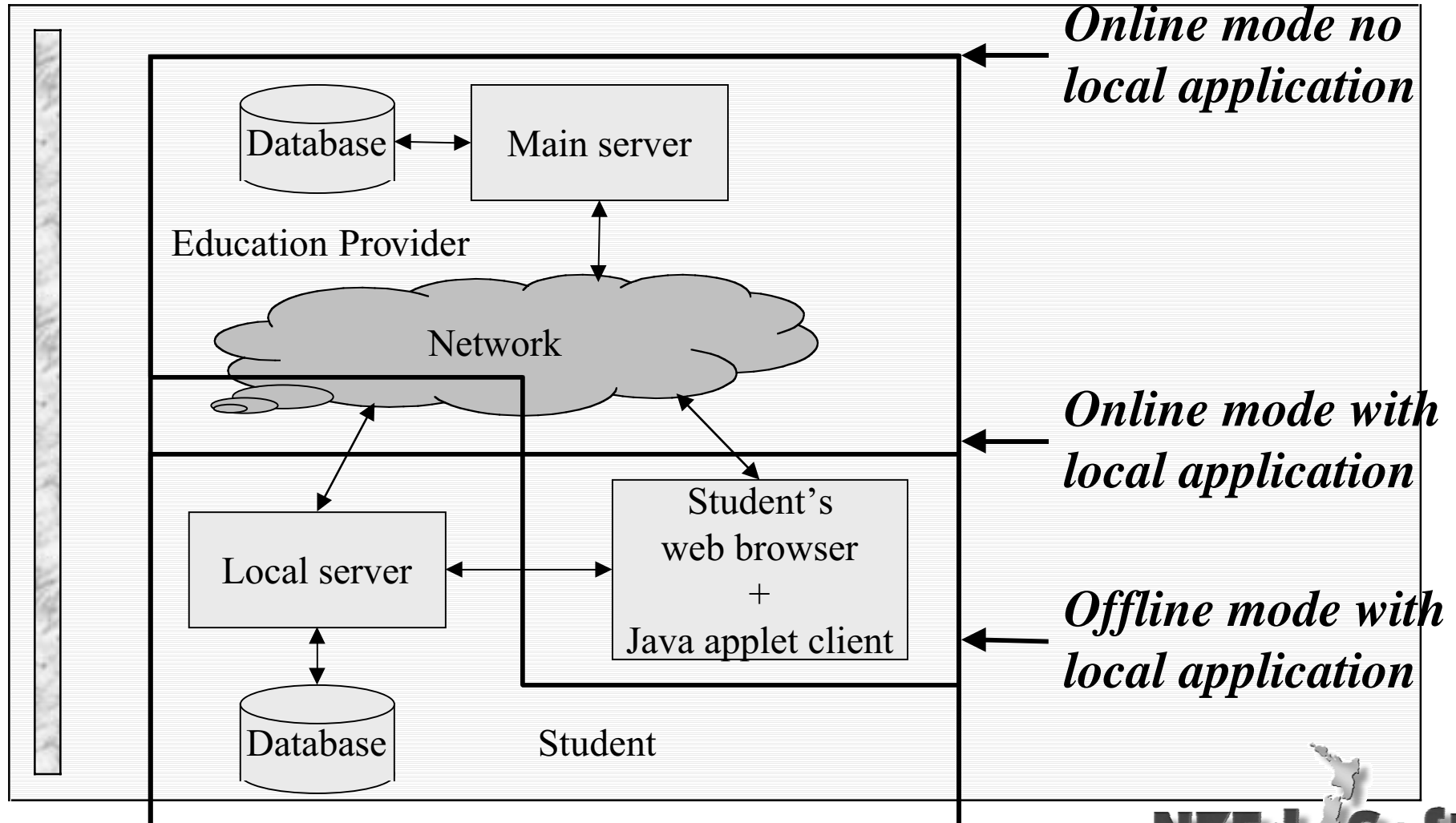
Tile uses hierarchy and partial order



Tools and standards

- ▼ The Tile system will provide easy-to-use tools to:
 - Configure the system - administrators
 - Edit structure, precedence and meta tags - teachers
 - Navigate structure and keep structured notes - students
- ▼ We will use open standards where possible
 - Software - Linux, Apache, SQL, JDBC, HTTP, XML
 - Meta-data - IEEE LOM, Dublin core

Tile architecture



Tile architecture - student interface

- Need a uniform interface for all modes
- ▼ Use web page & Java applet
 - for *online without* mode web access is the only viable solution - thus use it in all cases
 - applet is needed because we need complex application logic in HTML page for browsing and searching
 - applet can be downloaded with HTML page

Student side - why an application?

- Tile relies on an applet to navigate and view the educational material
- So why use a Java server application on the student's computer?
 - we need to access local disk for off-line browsing
 - reading media
 - logging student's access
 - writing profile information
 - Synchronising local database
 - Java applets can not access local disc due to security restrictions

Student side - Java security

- ✔ Applets loaded from a network may not:
 - read and write files on client file system
 - start programs, load libraries, define native method calls
 - make network connections *except to the originating host*
- ✔ There are two ways to trust an applet
 - obtain from local disc
 - being digitally signed by someone the user trusts
- ✔ No uniform treatment of either by browsers
 - application + applet avoids security issues
 - applet creates a socket to the originating application

Student side - the Tile solution

- ✓ The functional division
 - the applet presents the user interface
 - the *Tile server application* implements the database functions either locally, remotely or both
- ✓ *Online without application*
 - Java applet comes from remote server
 - applet talks to the remote server via a socket connection
 - the remote server maintains the database tables
- ✓ *Online with or Offline with application*
 - Java applet comes from the local server
 - applet talks to the local server via a socket connection
 - local and remote servers maintains database tables

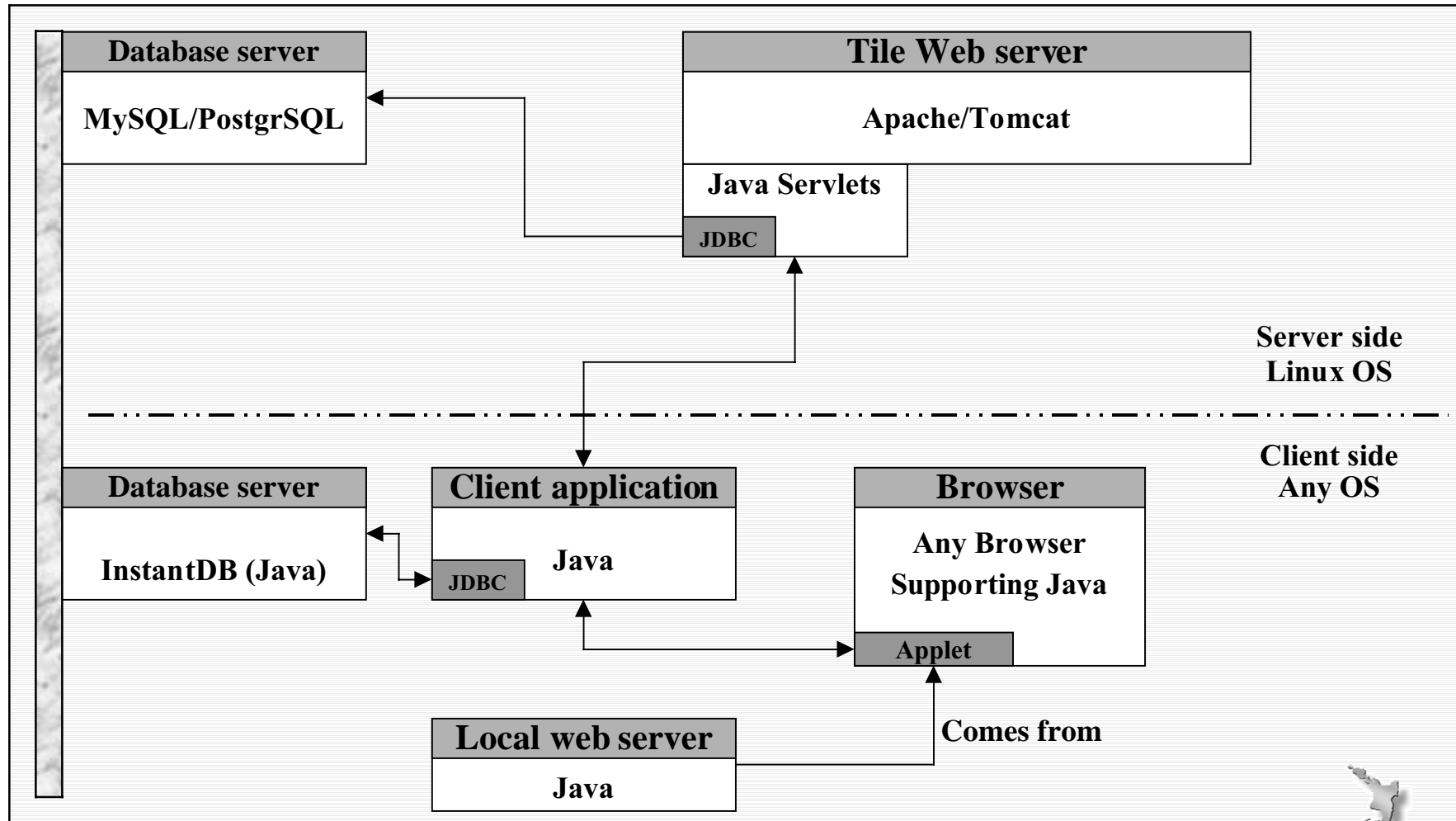
Server side - adding functionality

- We will use Linux, Apache, MySQL/PostgreSQL
- We must add functionality to the web server to provide access to the Tile database
- Adding functionality to the web servers can be achieved using a number of techniques:
 - CGI (Common Gateway Interface)
 - API (Application Programming Interface)
 - Using Java Servlets

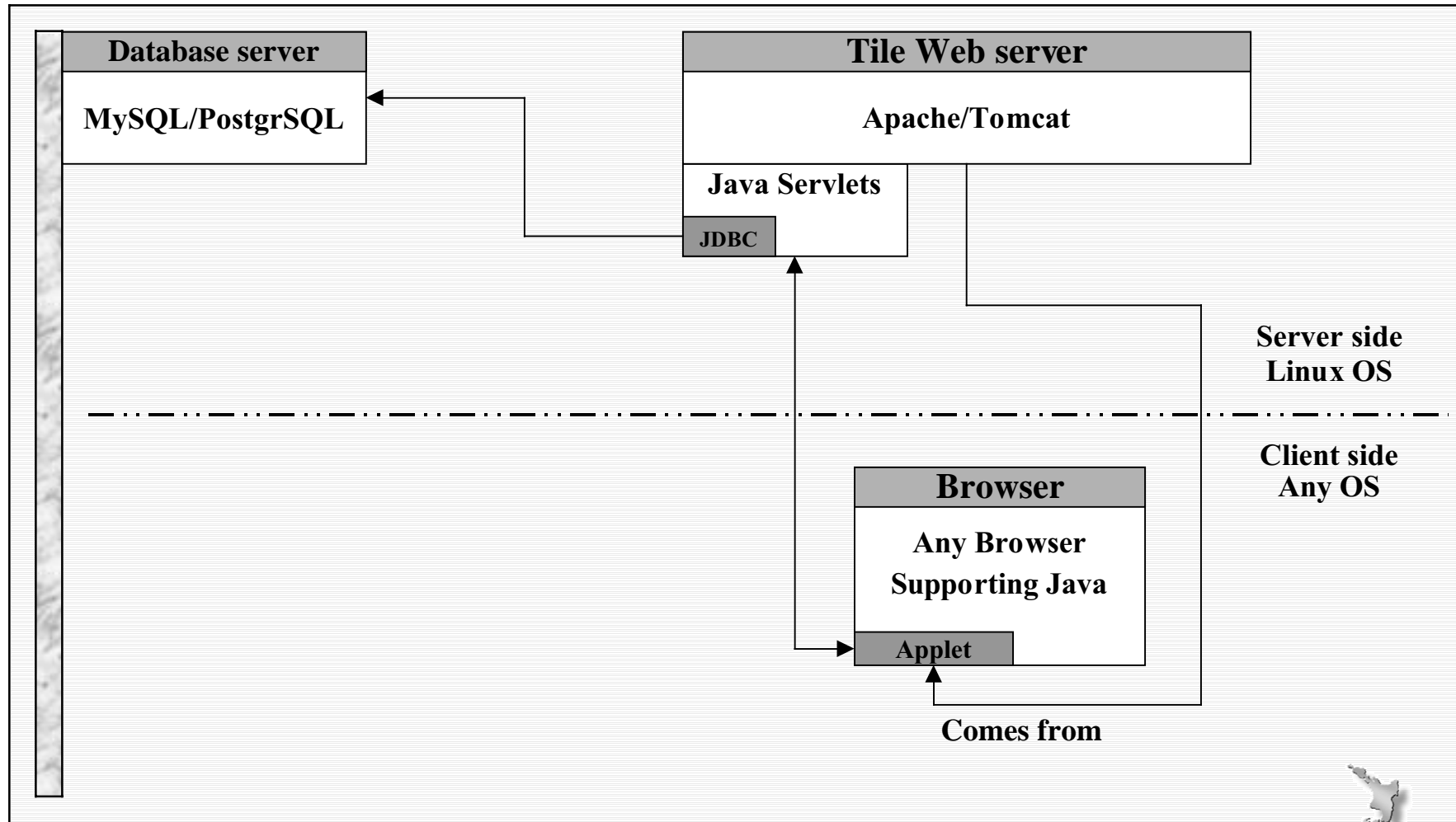
Server side - servlet approach

- We will use Java servlets
- Advantages are:
 - *performance* - like the API approach a single multi-threaded application can handle all user requests
 - *portability* - as they are written in Java servlets are platform-independent
 - *security* - Java security manager enforces a set of restrictions to protect resources on the server
- ✓ But web server must support a servlet API interface
 - Tomcat provides Apache with servlets

Tile - *with local application*



Tile - *without local application*



Results and further work

- Tile framework has now been defined and many components prototyped
- ▼ Other work ongoing includes
 - student modelling
 - AudioGraph multimedia tool development
 - querying via natural language
 - tools to assist high-level course design
- Questions?