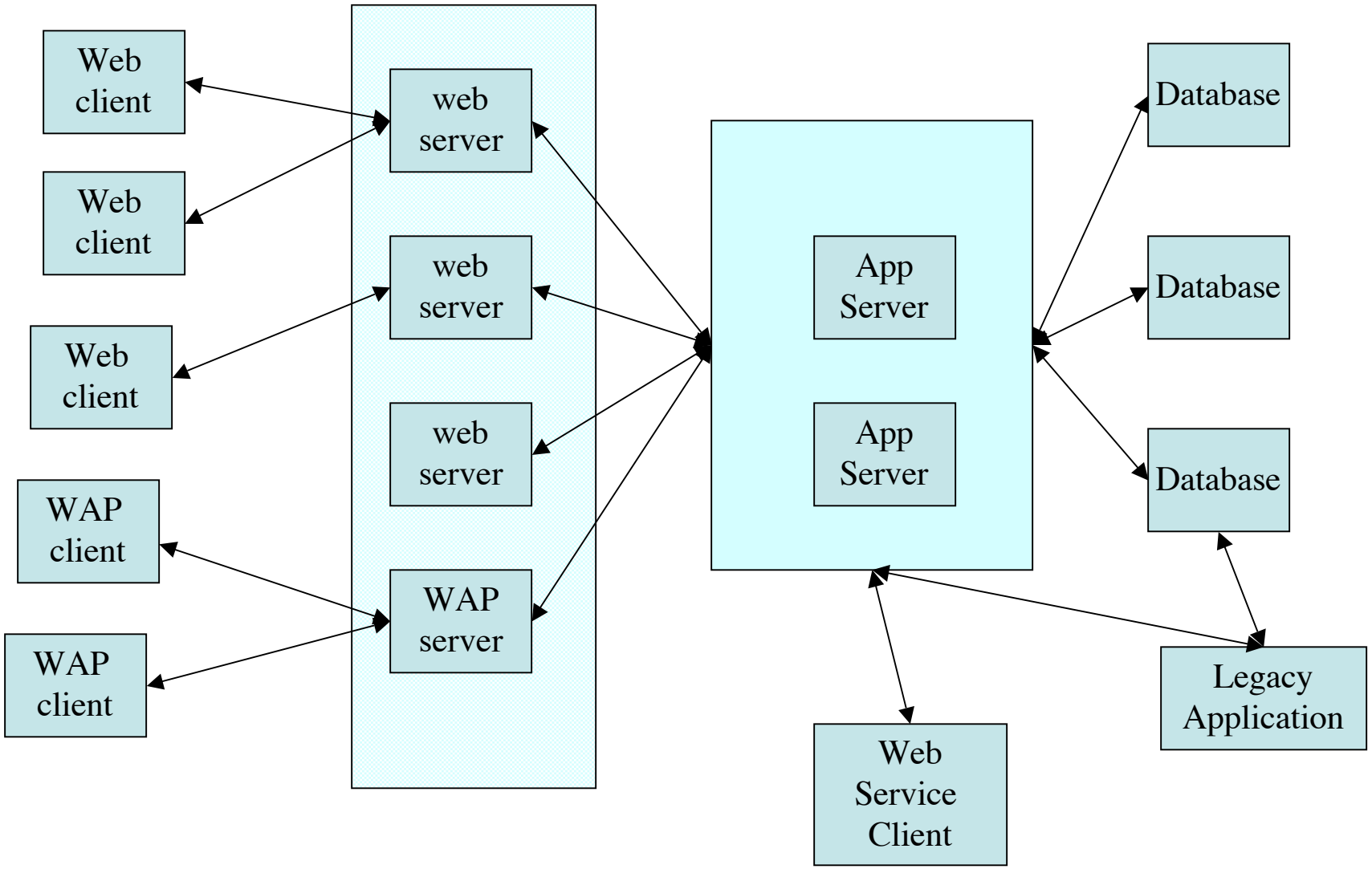
Enterprise Applications

William Pugh

Enterprise applications

- E-bay
- Amazon
- Campus course registration system

Architecture



Features of Enterprise Applications

- Persistent data
  - databases
  - integrity
  - transactions
- Interfaces
  - web clients
  - web services
  - legacy applications

More features

- Distribution and scaling
  - some enterprise applications may be deployed in low activity environment
  - should be possible to scale them if demand warrants it
  - often means distribution across multiple machines
  - often geographically distributed
- Resiliency to failure
  - one machine failure cannot shutdown an enterprise application

Databases

- Relational databases the standard mechanism
  - scales well to very large data sizes
  - handles transactions and failure well
- SQL the standard language for talking to them

JDBC

- Java standard for talking to databases
- Allows efficient mechanisms for talking SQL to databases
  - doesn’t just read/write text streams

Persistent Objects

- It is a pain to have to use SQL queries whenever you want to touch persistent state
- Better to have objects that reflect persistent state
  - and can check integrity rules and other business logic

Persistent Object solutions

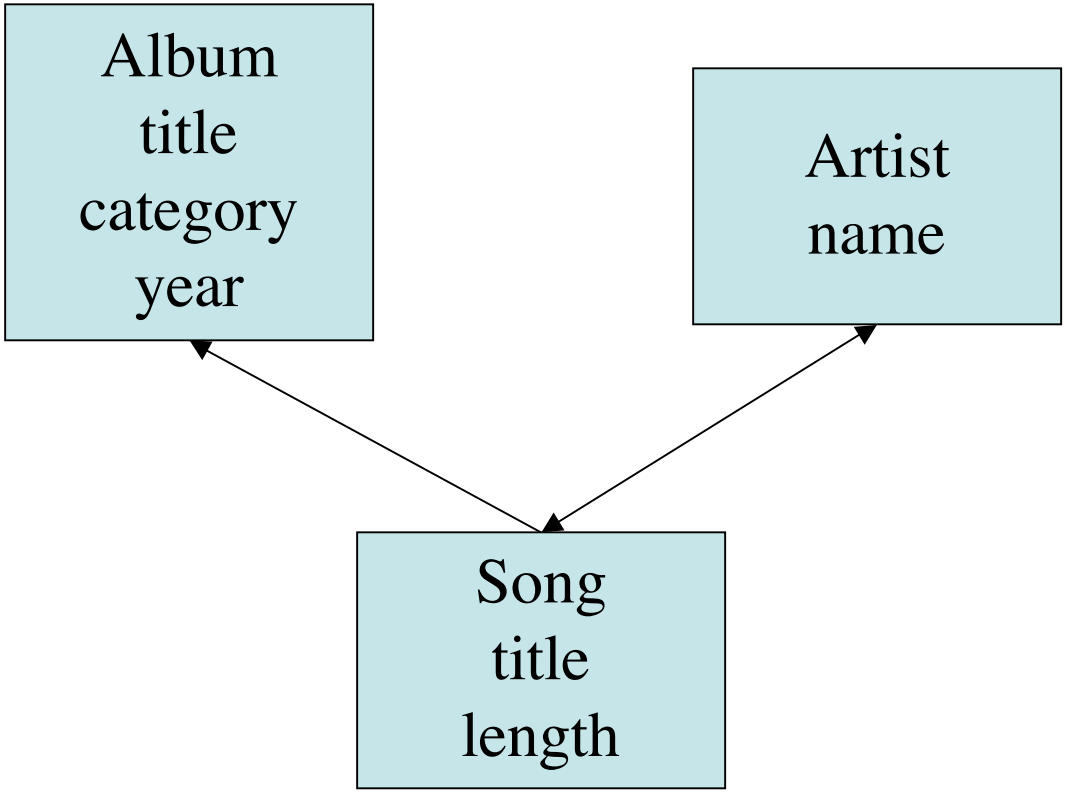
- EJB (Enterprise Java Beans)
  - BMP - Beans managed persistence
  - CMP - Container managed persistence
    - AppServer provides mapping between beans and database
- Java Data Objects
  - Good for small stuff
- Hibernate
  - an alternative to JDO

Databases in five slides

Transactions

- Atomicity - indivisible set of actions
- Consistency - respects database invariants
  - typically domain specific
- Isolation - ignorant of other, uncommitted transactions
- Durability - persistent, even in case of system failure

Song database



A song database

- Album:
  - *albumID*
  - title
  - category
  - year
- Artist:
  - *artistID*
  - name
- Song:
  - *songID*
  - albumID
  - title
  - length
- Performance:
  - songID
  - artistID

Locking

- Transaction 1 reads current balance
  - sees $100
- Transaction 2 reads current balance
  - sees $100
- Transaction 2 adds $10
  - sets current balance to $110
- Transaction 1 adds $10
  - sets current balance to $110

How to handle

- In threads, we would just use locks
- In databases, can use pessimistic or optimistic concurrency
  - pessimistic concurrency similar to standard locking
  - optimistic: don't lock
    - fail when committing if anyone else changed data you depended on

Web Presentations

- OK, it is great to have all this information available
- How do we present it over the web?

Servlets and JSP

- Servlets are an API for writing Java code that handles HTTP requests
  - provides easy access to all of the HTTP functionality (cookies, forms, …)
- JSP are a way that allows you to embed Java code inside a HTML document
  - compiled at the server, runs efficiently

Not the whole story

- extension mechanisms for JSP
  - just use special tags, no Java code, in your JSP
- static documents and images served via a standard web server
  - although a JSP server will do in a pinch
- Additional mechanisms to handle control flow and page composition
  - Struts, Java Server Faces, …

Rest of today

- Overview of Servlets
- Overview of JSPs