What are Web Services (WS)?
Self-contained, modular business applications that have open, Internet-oriented, standards-based interface (UDDI consortium)
Examples: online banking, e-commerce (Amazon.com, eBay), ordering and ticket purchase via WWW...

What is XML?
eXtensible Mark-up Language: these days, a de facto standard for exchanging and describing business data

How do WS transactions look like today?

Why divided functionality is bad?

- WS requests have to be transformed from an XML representation into the following representations:
  - the middleware’s internal one
  - the one of the programming language
  - the one of the database system
  and then back... This results in degradation in performance and makes the job of the developer much harder
- Divided functionality also leads to overlaps
- Divided functionality forces adding transaction processing and recovery components
- Divided functionality results in a plethora of competing WS specifications, many of them ignoring implementation aspects or relying on other specifications not worked out yet

We envision a system that
- is transparent to the end user
- is flexible enough to support a variety of applications
- scales well to support a large number of participants

The aims of the project
- To develop a flexible framework for defining specific transactional semantics
- To consolidate the disparate XML message management functionality of today’s middleware into a single native XML database system