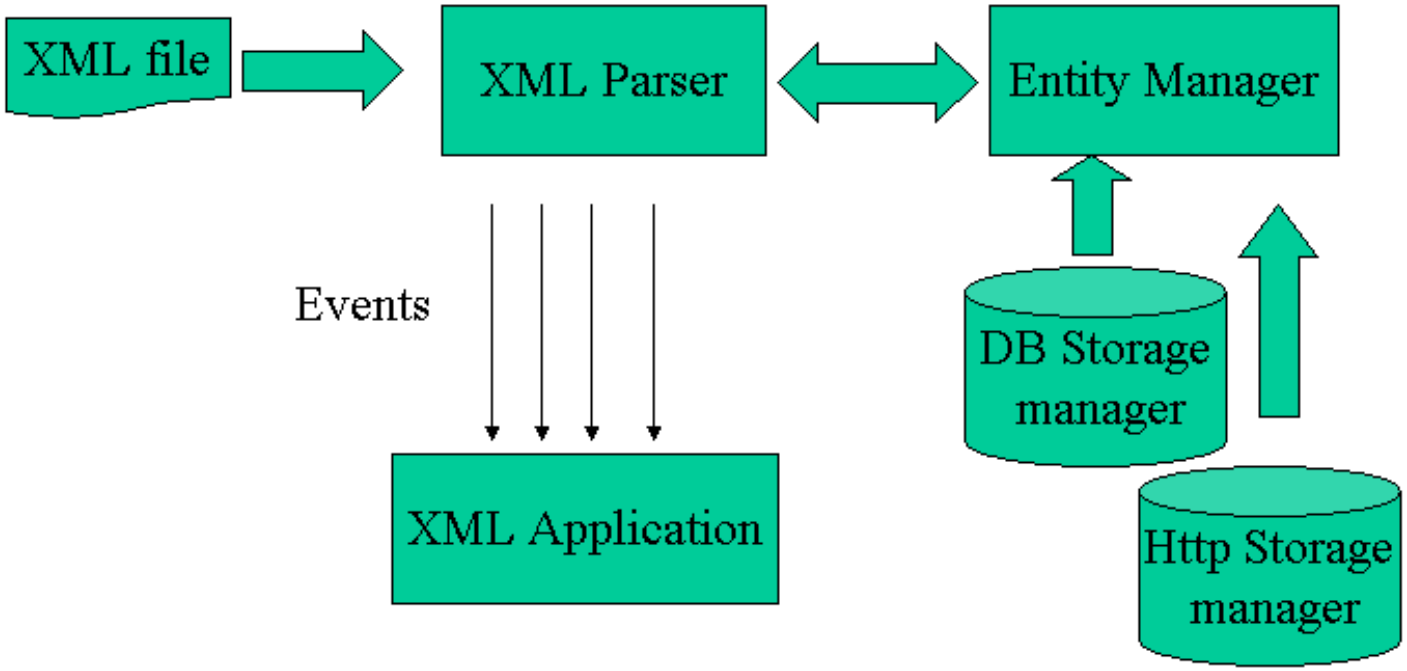
Steps to integrate XML

- How does XML processing work?
- Simple uses of passive DOM objects
- Adding behaviour to information
- A converter and translator subsystem
- The DNA of a large system
- The one and only requirement

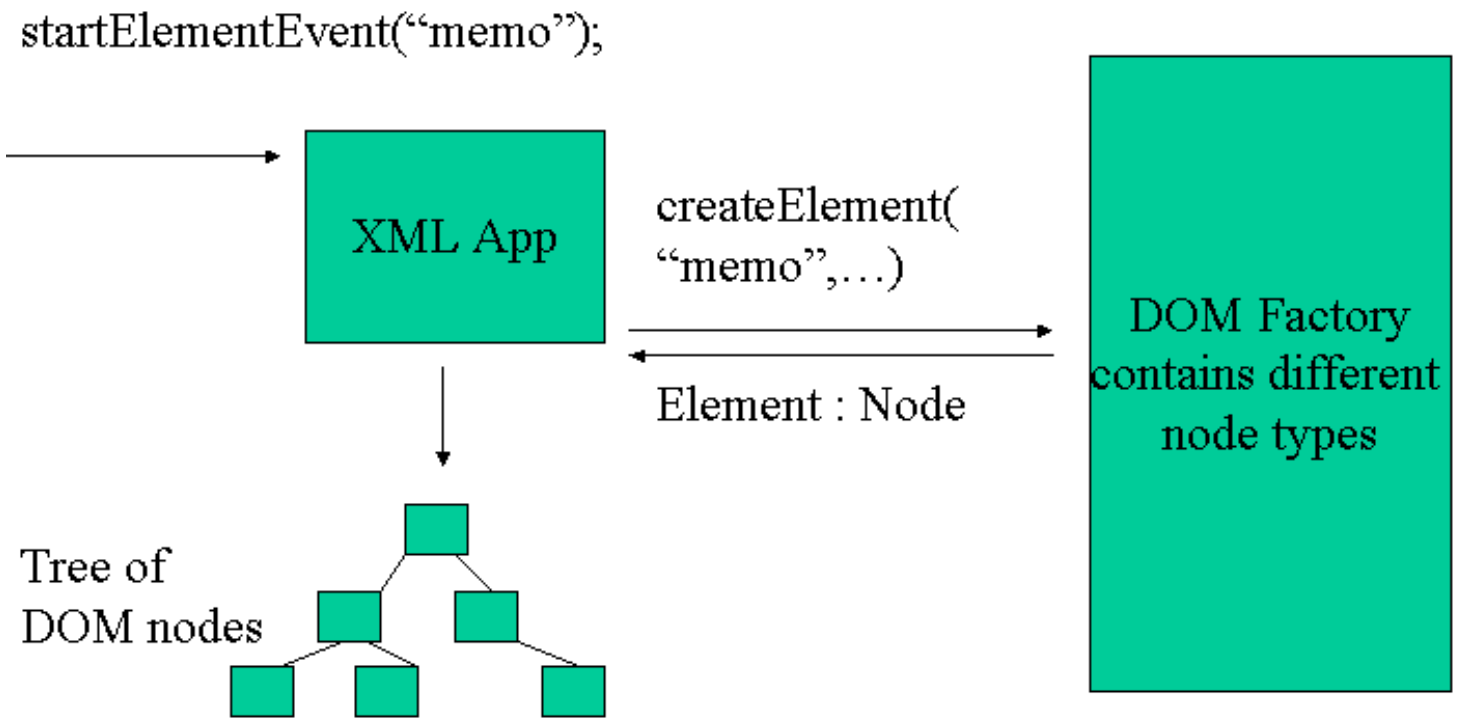
XML File to XML Application



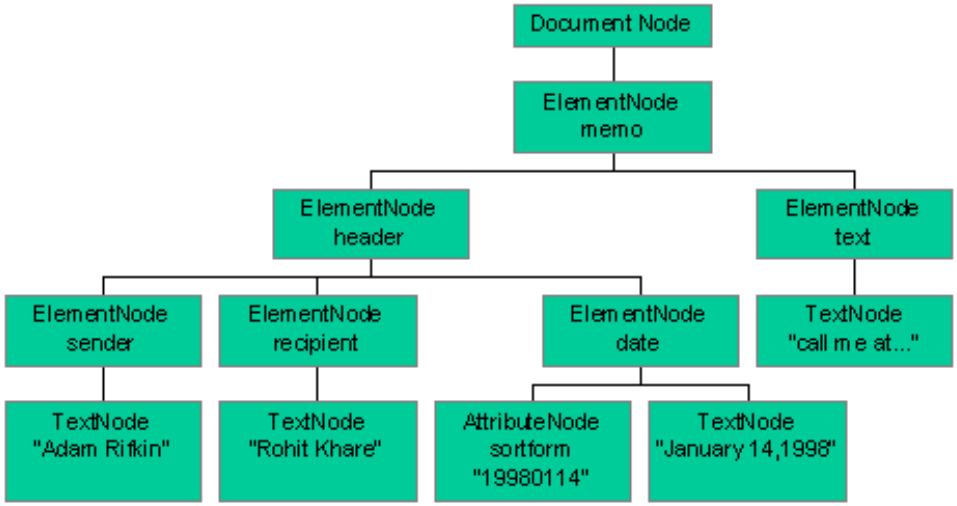
XML Application

- Can do on-the-fly processing
- Can build a tree of Document Object Model nodes in memory
- Can use a DOM Factory to get the nodes

XML Application to DOM



Simple memo.xml as DOM tree



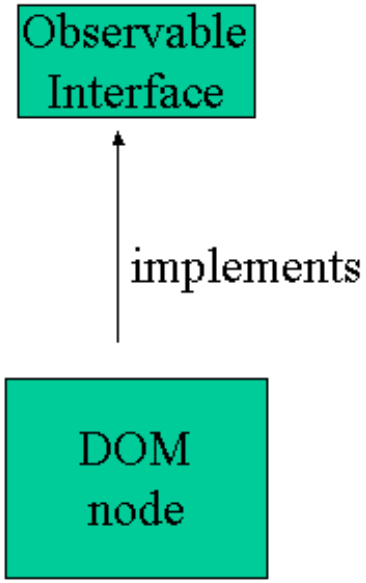
Simple uses of passive DOM objects

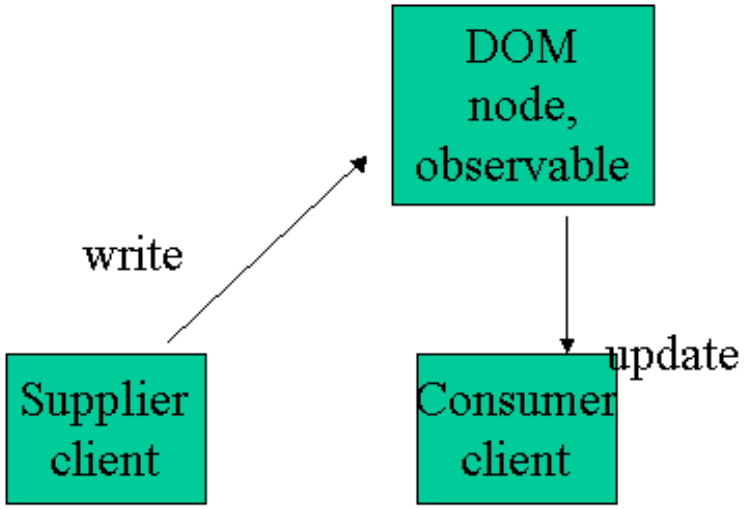
- System configuration
- Repository/meta information
- Debug and trace formats
- Semantic data streams
- composite message pattern

Advantages of XML information

- Code savings because only one format
- Standard tools can handle everything from config files to debug output, from editing to publishing
- Symbolic interfaces are more flexible but are still under control
- Information can evolve without code changes

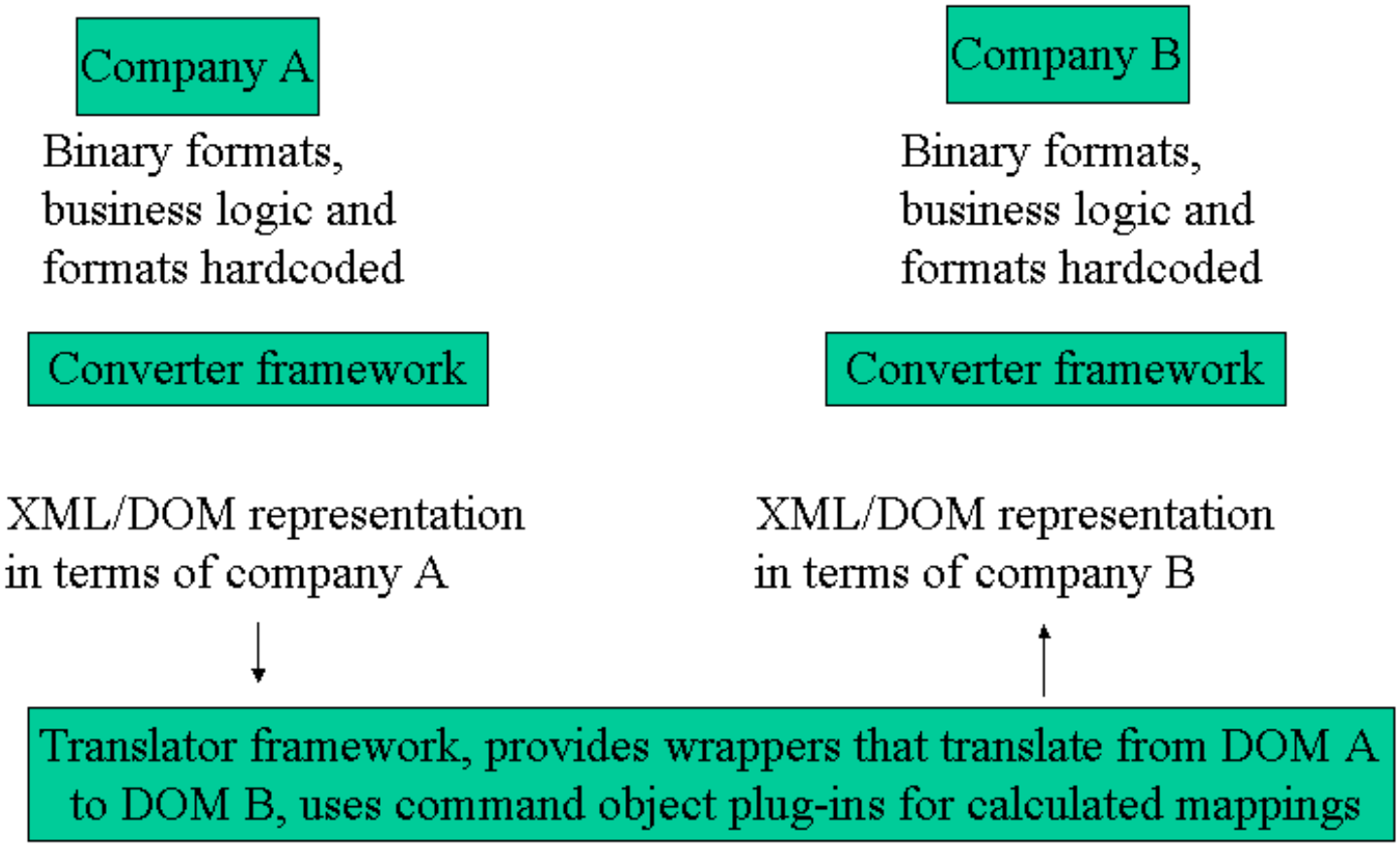
Adding behaviour to information





Simple blackboard implementation, enables information sharing and event driven processing without coupling clients

A converter and translator subsystem



The DNA of a large System

- Factories use XML information to build and assemble objects
- Objects use XML information to learn what they are supposed to do
- System behaviour emerges from cycles of transcribing XML into objects and those objects turning around and transcribing more XML "DNA" into further objects.

The one and only requirement

- Represent knowledge explicitly, in a human and machine readable form. This is true for application domain knowledge as well as for the processing system internals.