

FreeSB

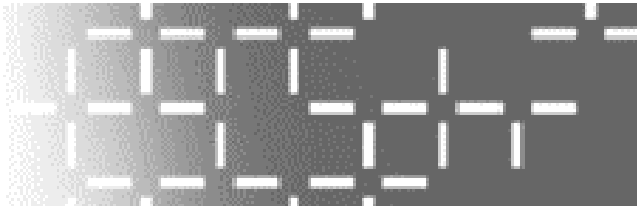
Get on the Bus!

Presentation Overview

- What IS an ESB?
- Why an Open Source ESB?
- Architectural Overview
- Key Features
- How it Works
- Traps for the Unwary
- Demonstration
- Questions

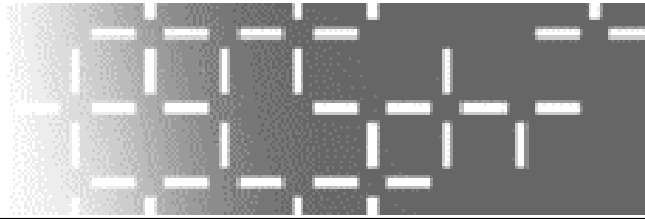
What IS an ESB?

- The term Enterprise Service Bus came into usage very late in 2002 and is still not clearly defined
- Common themes:
 - Distributed, decentralised messaging framework
 - Service-oriented middleware
 - Standards (XML/SOAP) based
 - Ideal for EAI
- Readily scalable by simply adding additional instances of a service



Why Use an Open Source ESB?

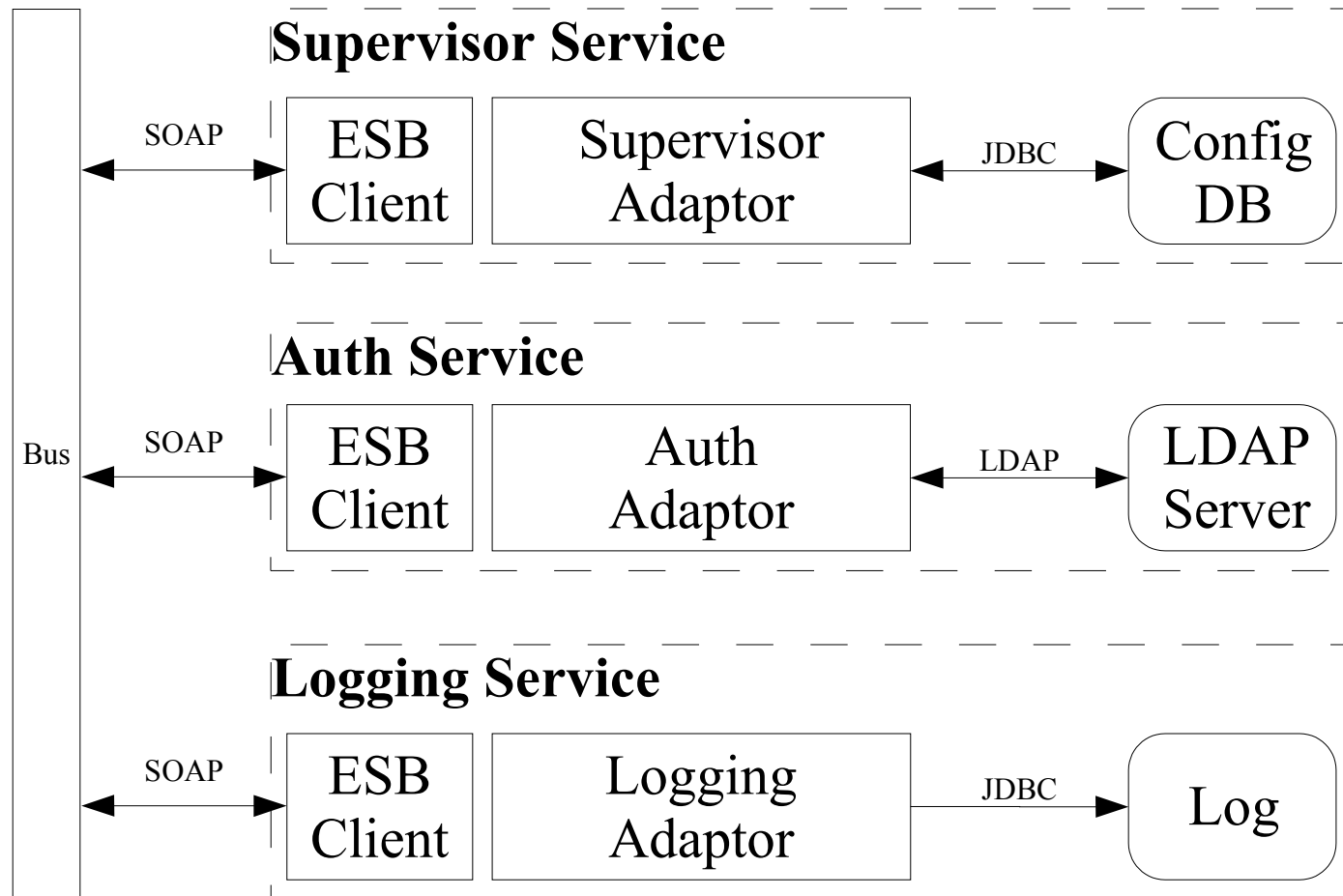
- Commercial products cost a lot of money
- Commercial products each provide only fixed methods of integration regardless of your needs
- Help develop a standard for ESBs and avoid vendor wars at the same time
- Service performance can be increased by adding arrays of cheap servers rather than having to upgrade monolithic servers

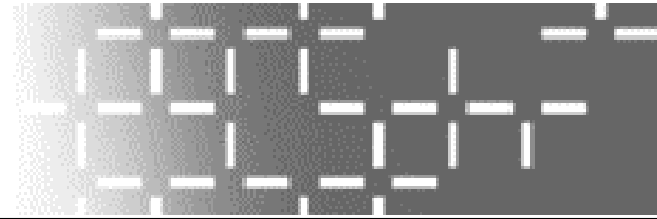


Why Build an Open Source ESB?

- We see ESBs as one of the prime technologies in the EAI toolbox
- With ESBs being predicted to be the “next big thing”, early adoption provides future opportunity
- Spherion sees a future in developing itself as a serious provider of Open Source solutions
- Spherion's core business is selling consulting services, not shrinkwrapped products

Architectural Overview



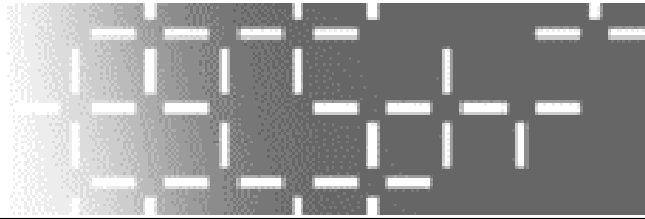


Key Features

- Customisable adaptor-based interaction with legacy systems to provide access to the ESB
- Horizontally and vertically scalable
- Logging, authentication, authorisation and management services
- Based on Open Standards
- The provided ESB client provides a convenient means of creating your own services around legacy applications

How it Works

- Apache Axis as transport engine
- Hibernate (non-JDO object persistence)
- Jakarta Tapestry (component based web framework)
- Also, see diagrams on the whiteboard



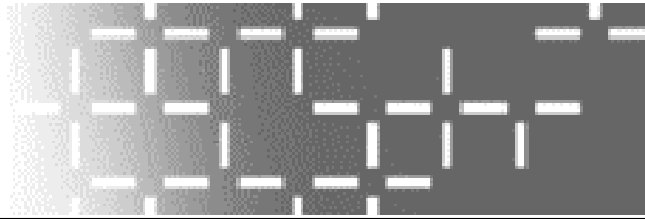
Security

- Principals used to track sessions
- Role based
- Permissions
- MOANs
- Container-level (HTTPS, BASIC AUTH)

Traps for the Unwary

- Apache Axis (reliability, exception handling)
- JMS implementation
- Hibernate / Persistence (compatibility)
- SOAP performance (HTTP serialization etc.)

Demonstration



Questions?