



**Poseidon House
Castle Park
Cambridge CB3 0RD
United Kingdom**

TELEPHONE:
INTERNATIONAL:
FAX:
E-MAIL:

**Cambridge (01223) 515010
+44 1223 515010
+44 1223 359779
apm@ansa.co.uk**

ANSA Phase III

Topaz: A Structured Web Service Architecture

Mark Madsen

Abstract

Design and development of webservers must take requirements associated with future extensibility into account. Object technologies have a proven record in the area of large-scale system design. They are well adapted to extensibility requirements, especially when the required additions of functionality cannot be foreseen at design time. Secondly, they provide a basis for permanent in-service re-engineering of the support components, taking advantage of new capabilities and product deliveries.

Topaz demonstrates how use of object technology simplifies the design of web based services. Topaz builds on existing webserver technology using current CGI methods and techniques.

Implementation of Topaz is scheduled to start at the beginning of April 1996.

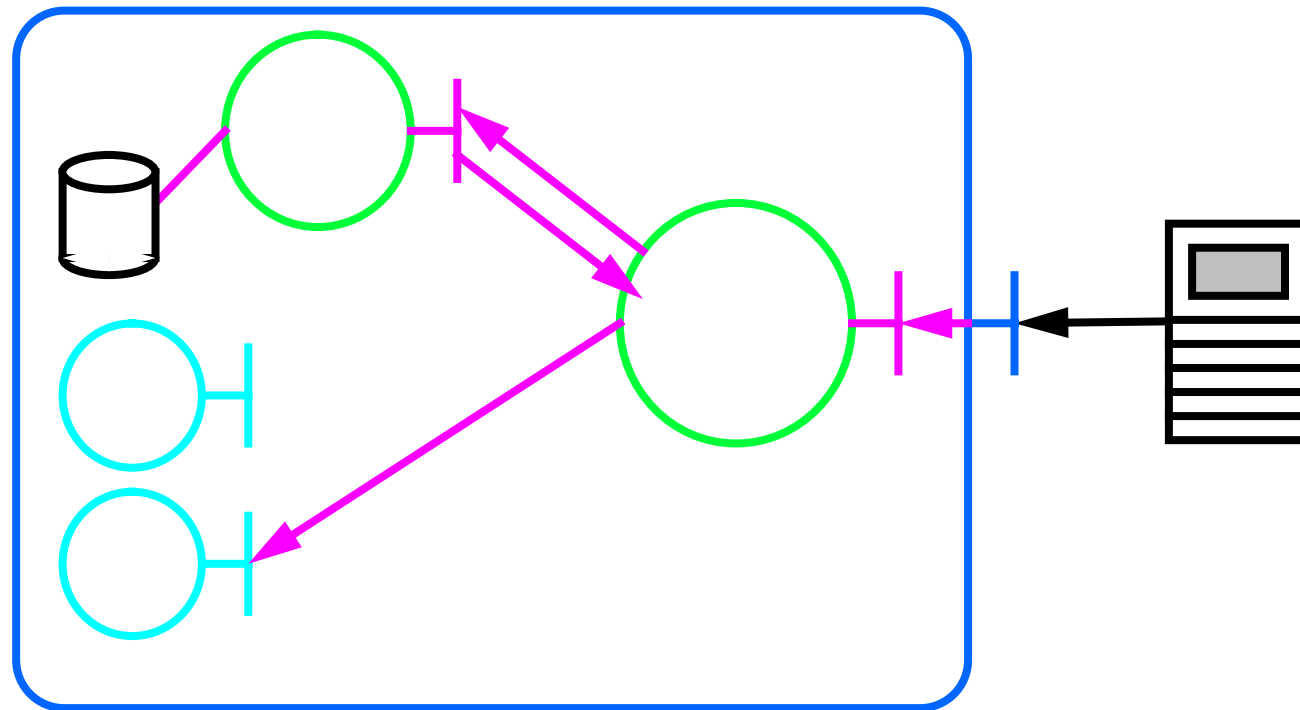
APM.1715.01

Approved
Briefing Note

23rd May 1996

Distribution:
Supersedes:
Superseded by:

Topaz: A Structured Web Service Architecture

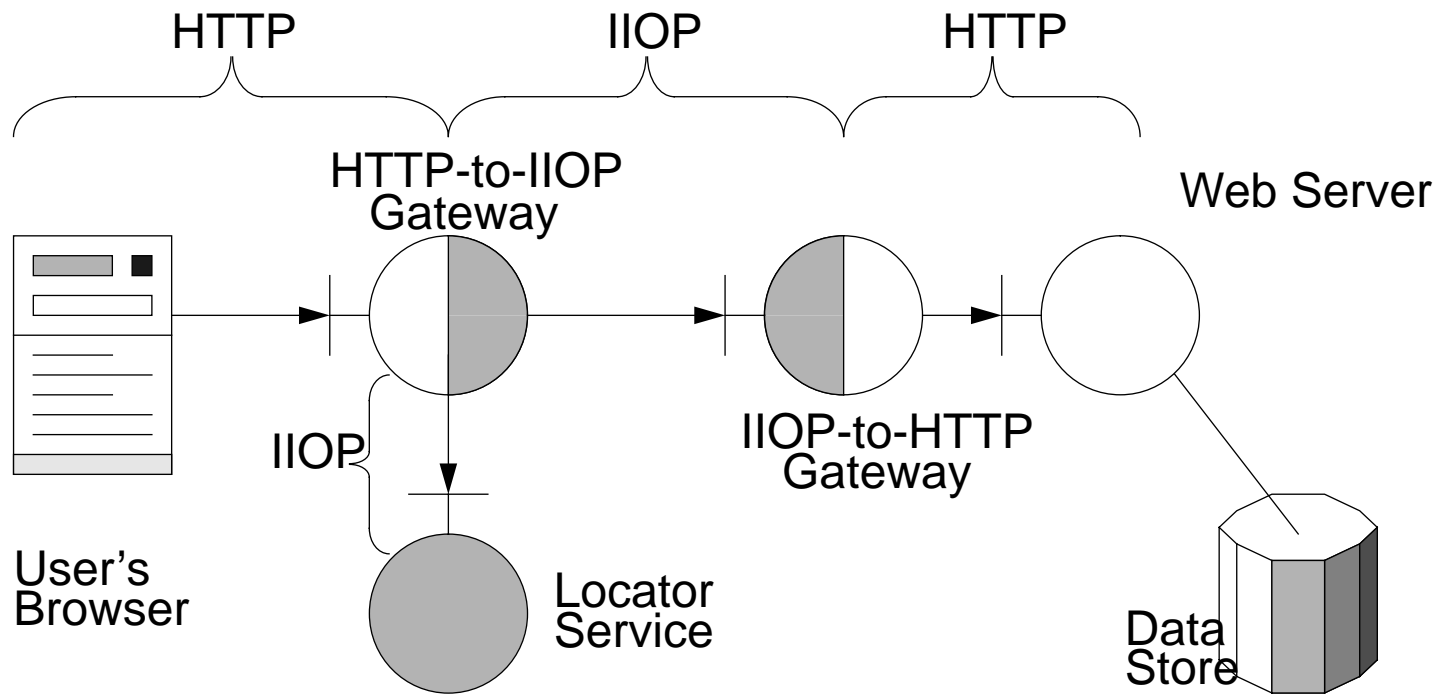


Situation Overview

- **ANSAweb Project showed how object technology can integrate CORBA services with the WWW**
- **ANSAweb based on idea of structuring interactions for interoperability on the WWW**
- **Webserver technology limitations responsible for most functional limitations on the WWW**

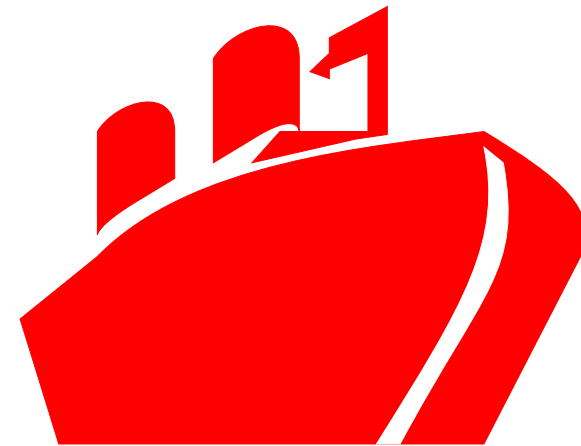


Background: ANSAweb Structure



Business Risks

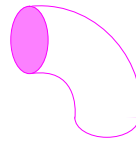
- **Server technology gap will widen as need to deliver live services increases**
- **Servers will become**
 - **harder to maintain**
 - **more expensive**
 - **larger and monolithic**
- **Server technology will fall into a single-company stranglehold**



Central Issue

Building Better Web Services

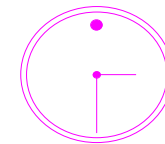
flexibly



cheaply



rapidly



Development Options

- **Extend the capabilities of the existing CGI mechanisms**
 - merge available toolkits
 - involves joining incoherent technologies
- **Develop webservers with a new service architecture**
 - structured
 - distributed
 - object-based

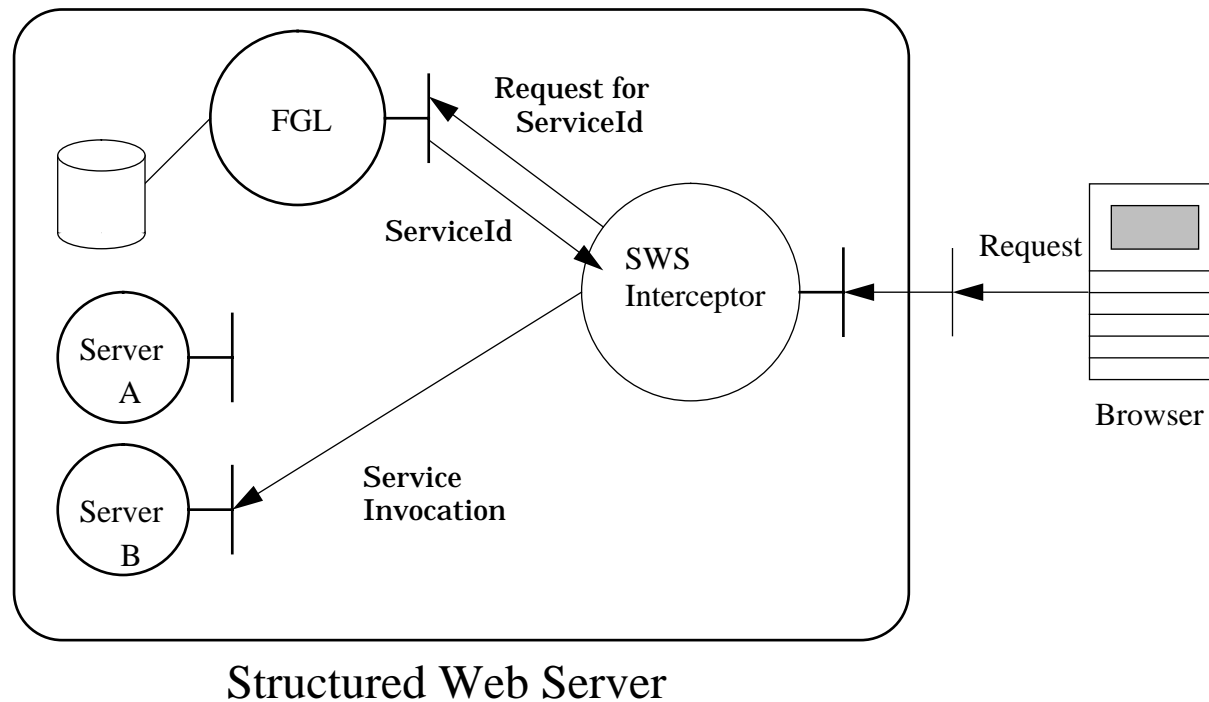


Recommendation

- **Implement the Topaz structured webserver**
- **Features**
 - most of the required technology has been built under ANSA
- **Advantages**
 - proposed architecture is new, better than existing servers
 - compatible with present CGI mechanism
- **Benefits**
 - simple integration with backend CORBA services and legacies
 - true distributed services allow new types of application



Development Approach



SWS is the Structured WebServer.

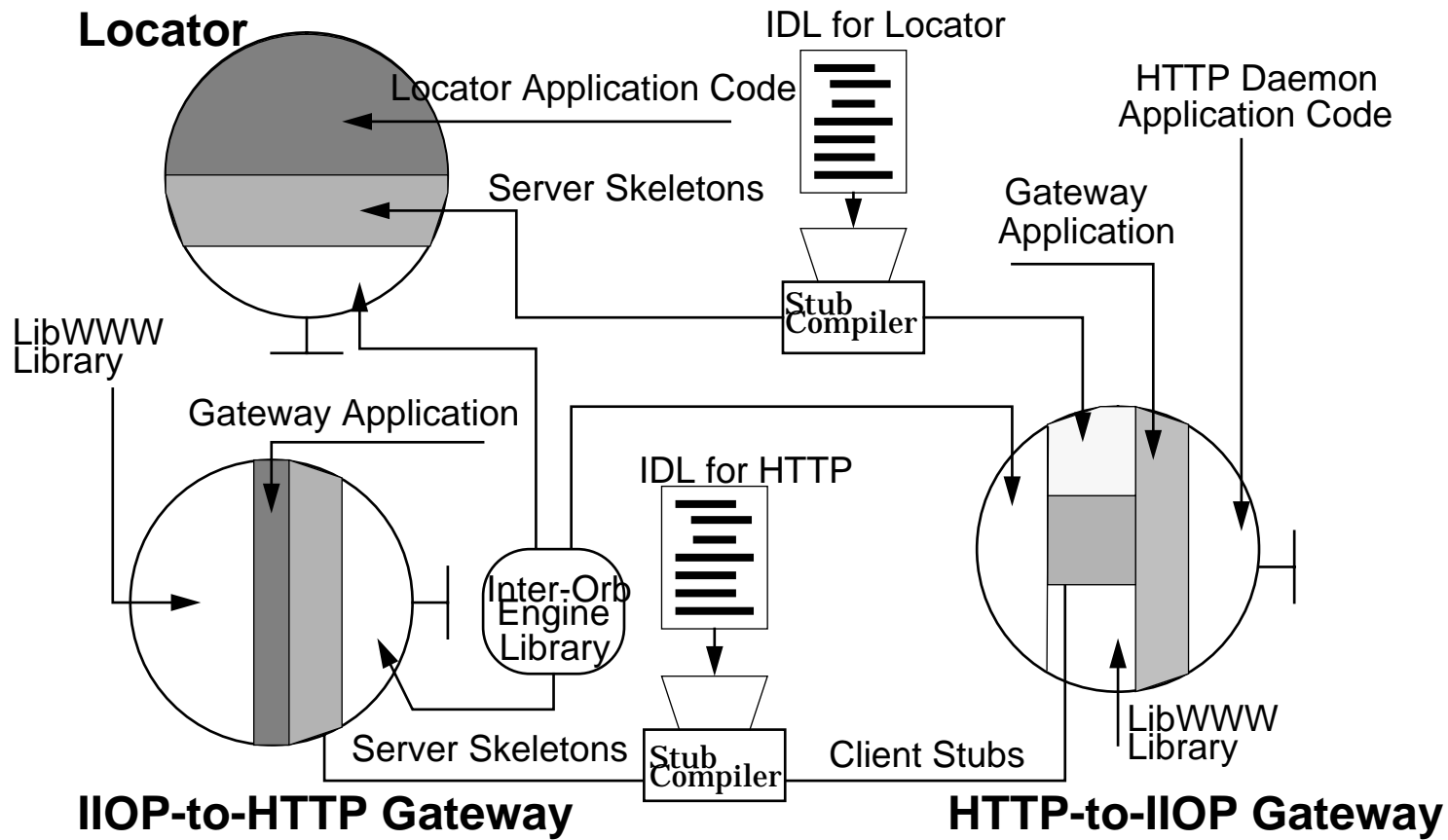
The Interceptor is the component that handles incoming requests and diverts them to the appropriate backend service.

Here FGL is the Fine-Grained Locator. This is based on the locator used in ANSAweb, but is finer-grained in locating objects at the level of individual services.

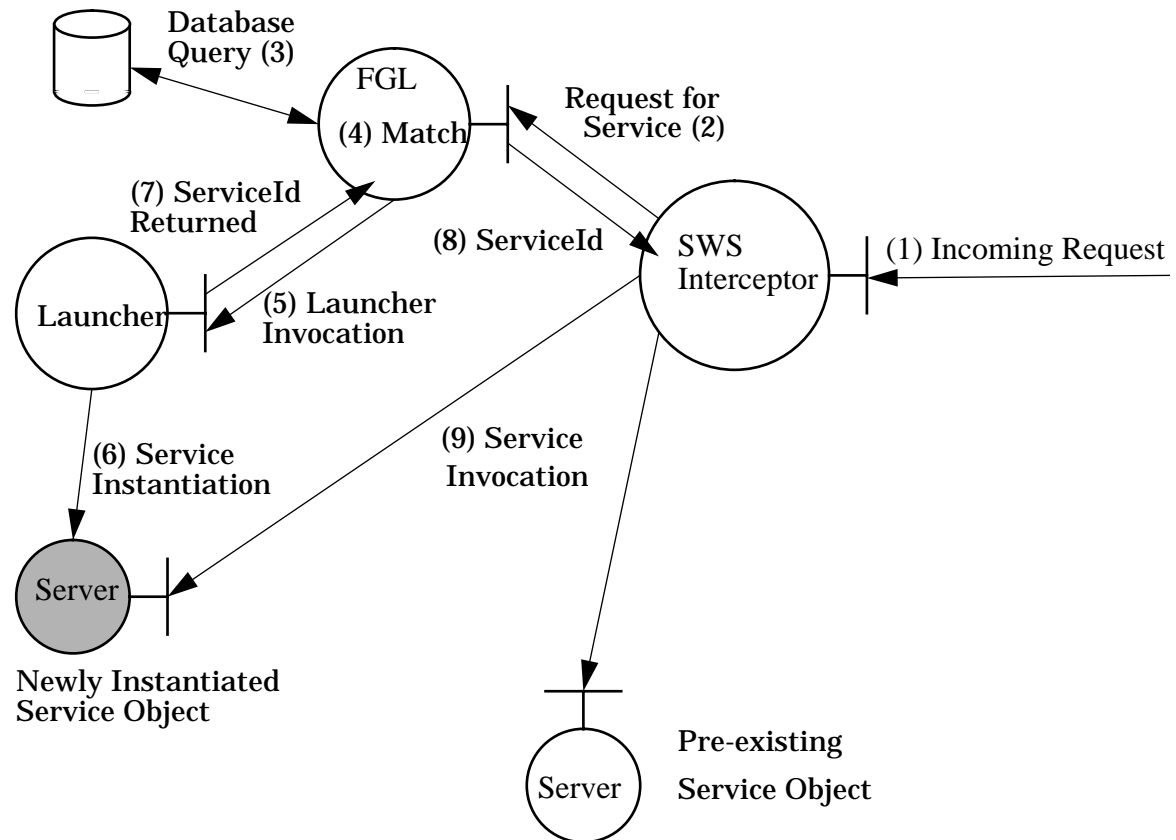
- Basic 3-tier architecture is shown in the above diagram



ANSAweb Locator and Gateway Technology



Topaz Operations Processing



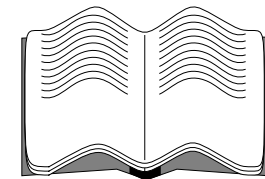
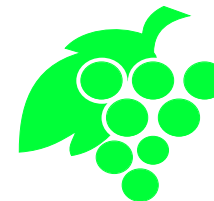
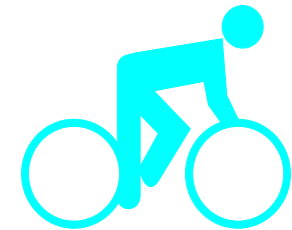
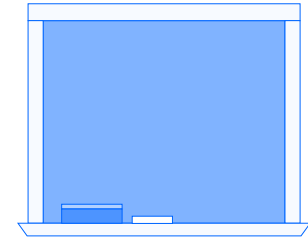
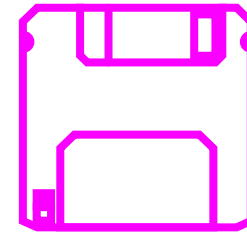
Benefits

- **Webserver interoperability mechanisms that can be used to leverage existing services built on CORBA.**
- **Simple extensibility/upgradeability for the server developer and maintainer.**
- **Better performance characteristics through using object mechanisms for service instantiation and load balancing.**
- **Compatibility with existing standard web server systems.**
- **Basic object support mechanisms can be implemented in CGI and further object services added on to a single CGI middle tier.**
- **A testbed for the Java technology developed within Jade, and for other projects delivering live services or mobile code.**



Deliverables

- Design presentation and document
- Launcher program and code
- Locator program and code
- Demonstrator application
- Performance testing
- Technology transfer presentation
- Final project document and summary.
- Paper submission for WWW6 conference.



Application Examples

- **Server load balancing scenario**
 - server front end/interceptor observes saturation of a service
 - new service object instantiated to spread load
 - interceptor shares new requests between objects automatically
- **Remote service invocation scenario**
 - locator maintains service information on remote services
 - locator passes incoming service requests to external services
 - client can use (invisible) services supplied by a third party



Timescales

- **First month**
 - design document and presentation
 - fine grained locator
- **Second month**
 - launcher
 - demonstrator application
- **Third month**
 - performance testing
 - technology transfer presentation
- **Fourth month**
 - final project document and summary
 - paper submission to WWW6 conference

