

TRW

Inter-operable Server Smalltalk and Client Java Architectures

Presentation to European Smalltalk
User Group Summer School

Brescia, Italy
August 31, 1998

Kirk D. Blackburn
Kirk@bdmtech.com

Greg Frohring
gfrohrin@bdmtech.com

Gregory P. Smith
gsmith@bdmtech.com

Commercial Information
Technology Group

Structure of Presentation

- Business Needs
- Architectural Givens
- Five Client-Server CORBA Architectures
- Benefits/Shortcomings
- Why Server Smalltalk?
- Summary/Conclusions

Business Needs

- System to Provide:
 - Multi-Media Content Creation
 - Web-based Content Delivery
 - Enable “Just In Time” Training
 - Central Administration of Content
 - Object-Oriented Database Storage/Retrieval
 - Scalable to Hundreds of Thousands of Users



Architectural Givens

 Netscape

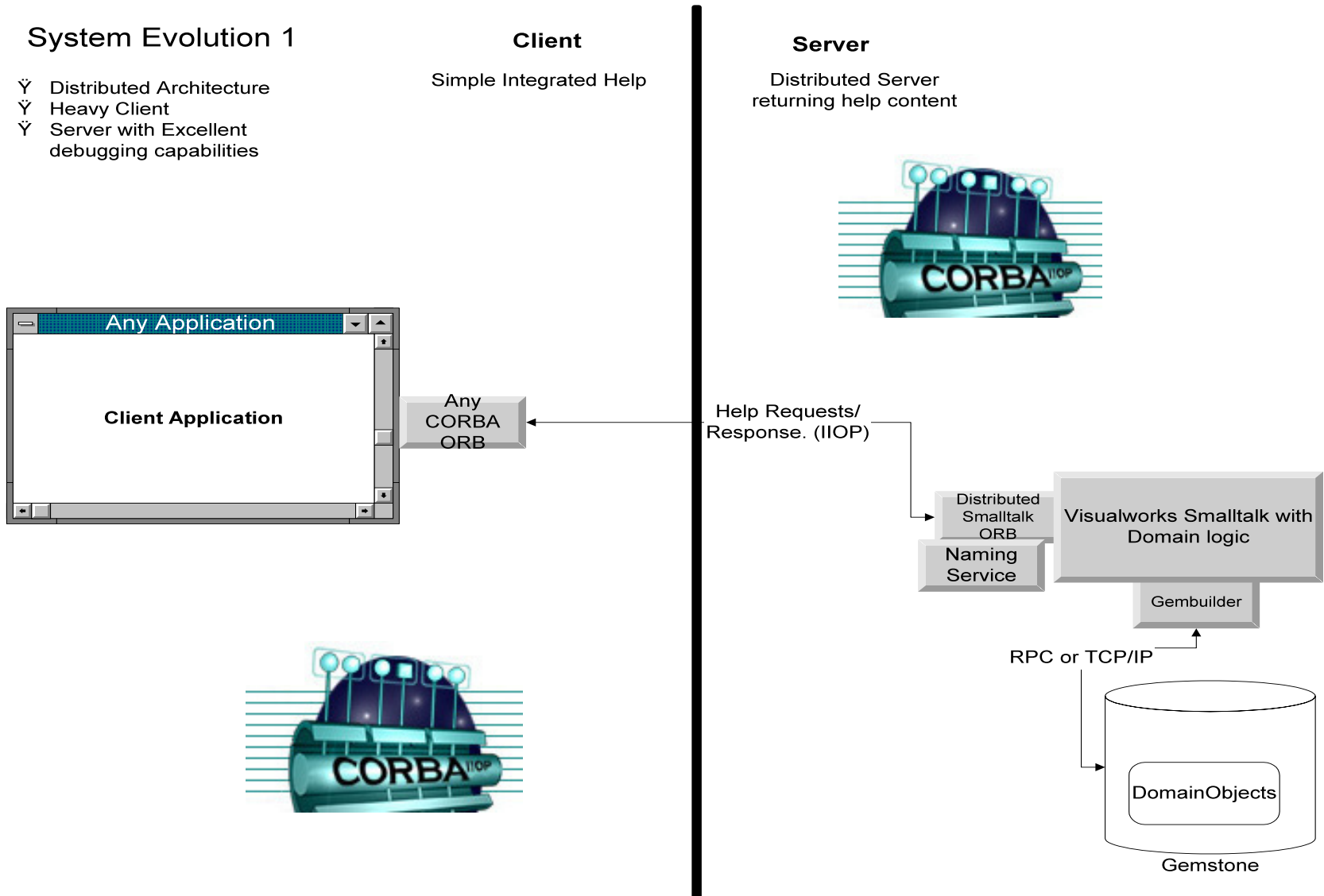
- Web Clients (Netscape, IE)
- Java Clients (Applications /Applets)
- Smalltalk Clients (VisualWorks Applications)
- CORBA (Multiple ORBs)
- Server Smalltalk (VisualWorks)
- Gemstone/S Application Server/DB



Architecture I

System Evolution 1

- ÿ Distributed Architecture
- ÿ Heavy Client
- ÿ Server with Excellent debugging capabilities



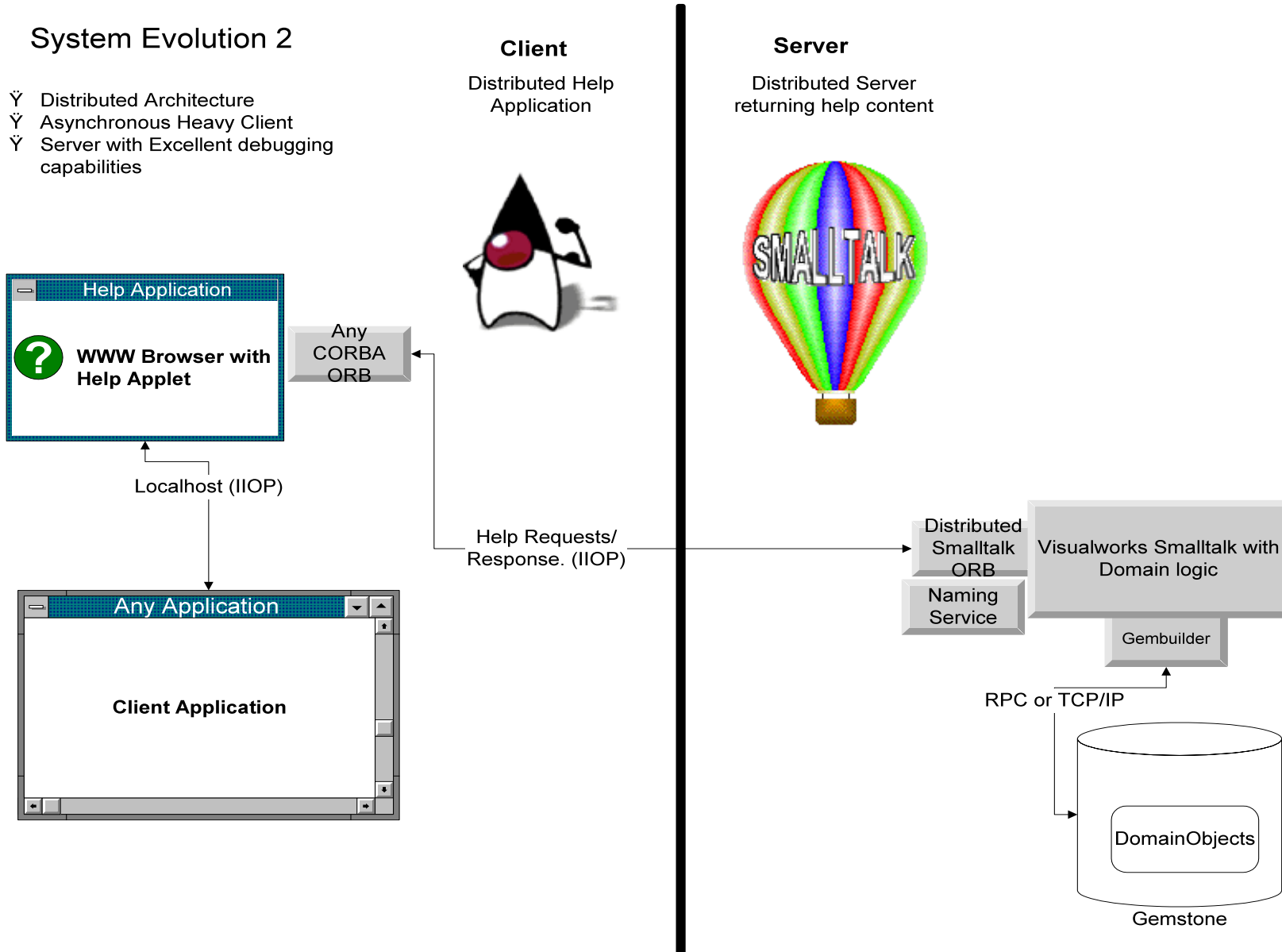
Architecture I

- Pros:
 - Uses CORBA for Remote Messaging
 - No Webserver Required
 - Heterogeneous Language Integration
 - It “worked” (J. Bostrom quote)
- Cons:
 - Uses IIOP for Data Transfer
 - Heavy Synchronous Client
 - COS Lifecycle/Timeout Issues
 - Non-optimal Memory Utilization
 - Required Application Source Modifications

Architecture II

System Evolution 2

- Y Distributed Architecture
- Y Asynchronous Heavy Client
- Y Server with Excellent debugging capabilities



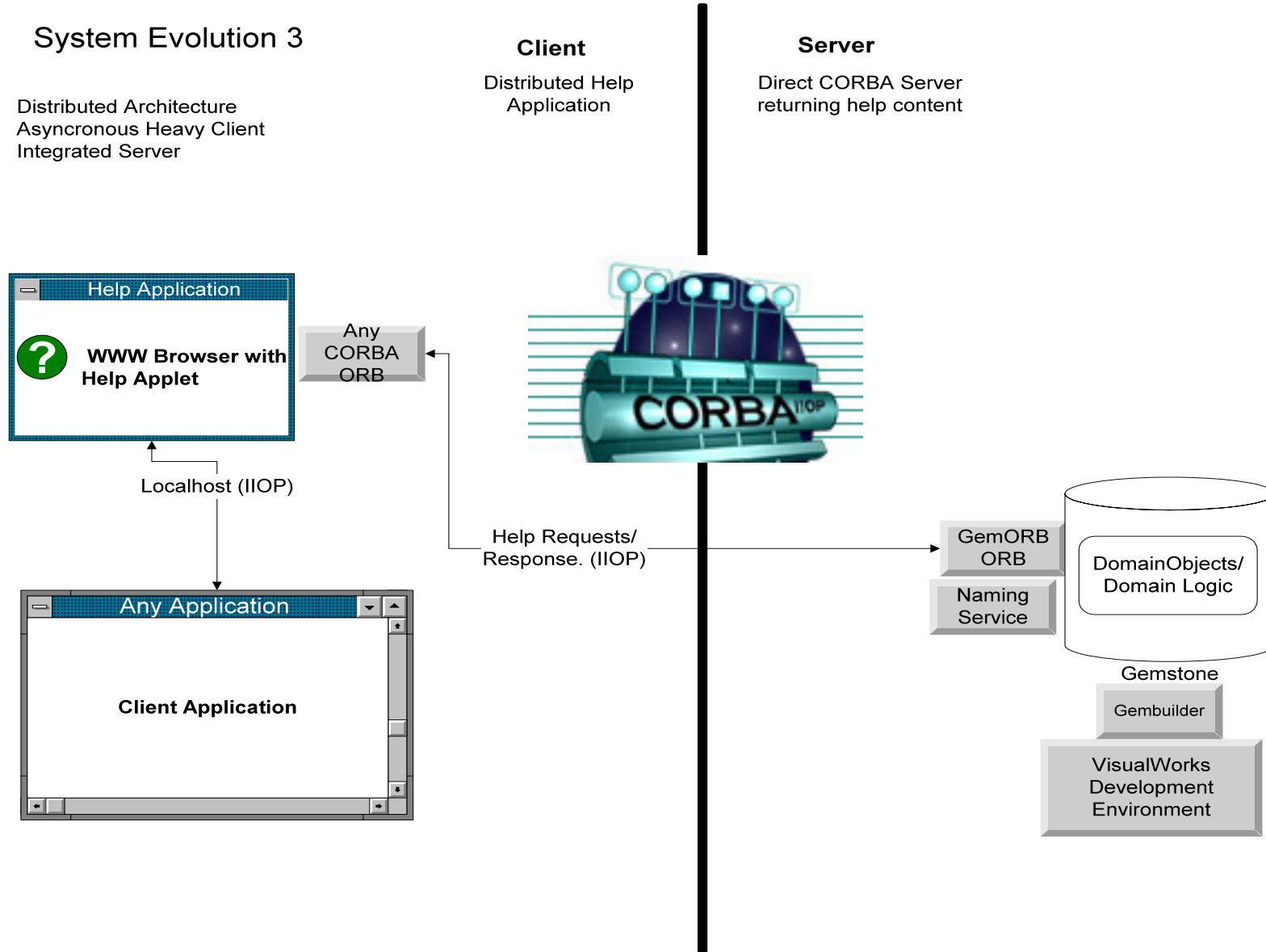
Architecture II

- Pros:
 - Uses CORBA for Remote Messaging
 - No Webserver Required
 - Asynchronous Client Communication (non-blocking)
 - Minimal Application Source Modifications
- Cons:
 - Uses IIOP for Data Transfer
 - Heavy Client (Content Generation Logic)
 - Lifecycle/Timeout Issues
 - Non-optimal Memory Utilization
 - Instability of Large Applets Running in Browser VMs

Architecture III

System Evolution 3

- Y Distributed Architecture
- Y Asynchronous Heavy Client
- Y Integrated Server



Architecture III

- Pros:
 - Uses CORBA for Remote Messaging
 - No Webserver Required
 - Asynchronous Client
 - COS Lifecycle/Timeout Issues Minimized
 - Memory Requirements Optimized
 - Retains VisualWorks Development Environment
 - Gemstone Runtime Environment (Scalability, Partitioning, Native Threads)
- Cons:
 - Uses IIOP for Data Transfer
 - Heavy Client (Content Generation Logic)
 - Loss of Robust VisualWorks Engine for Debugging
 - Instability of Large Applets Running in Browser VMs

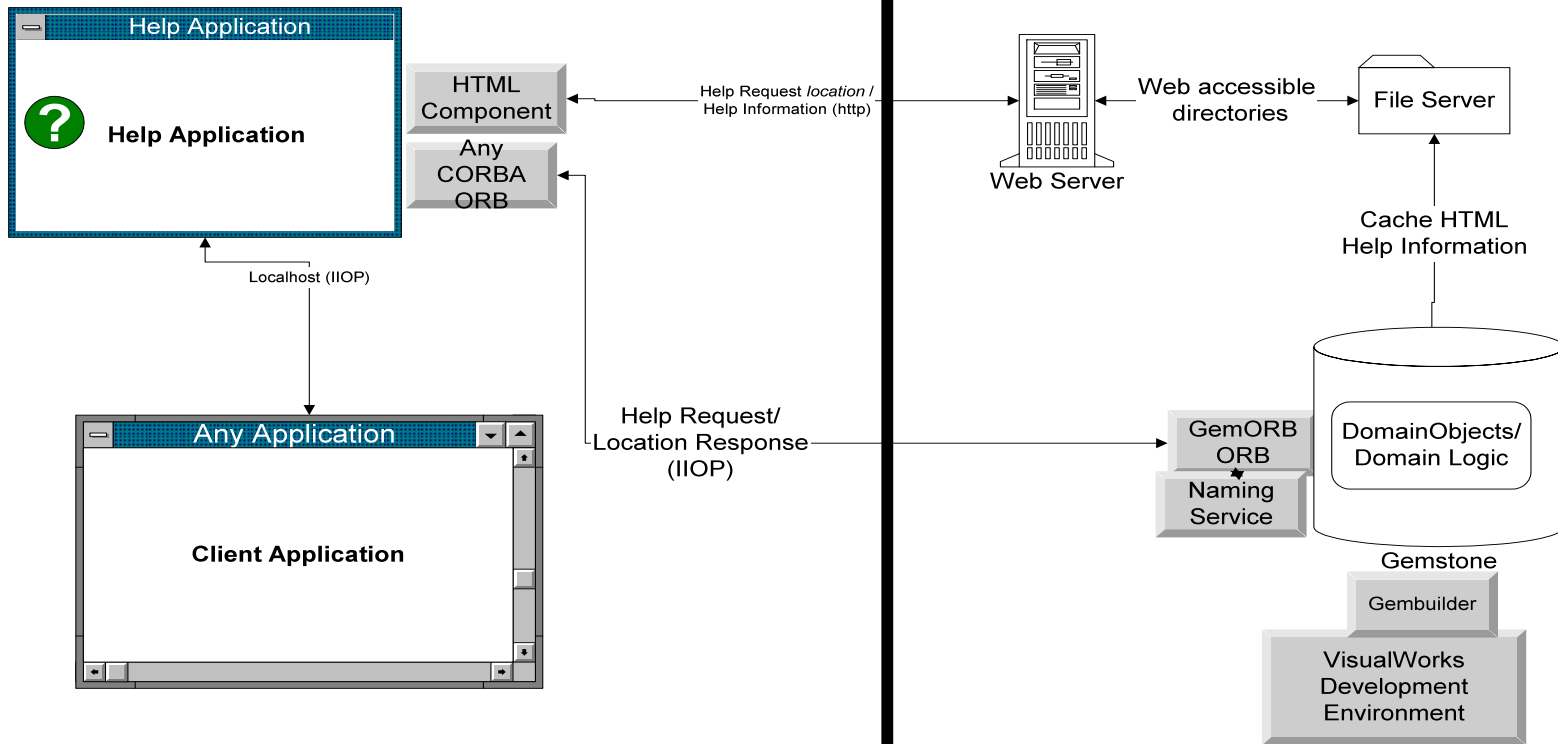
Architecture IV

System Evolution 4

- Y Distributed Architecture
- Y Asynchronous Thinner Client
- Y Integrated Modular Server

Client
Distributed Help
Application with
Web Browser Capabilities

Server
Direct CORBA Server
returning help location



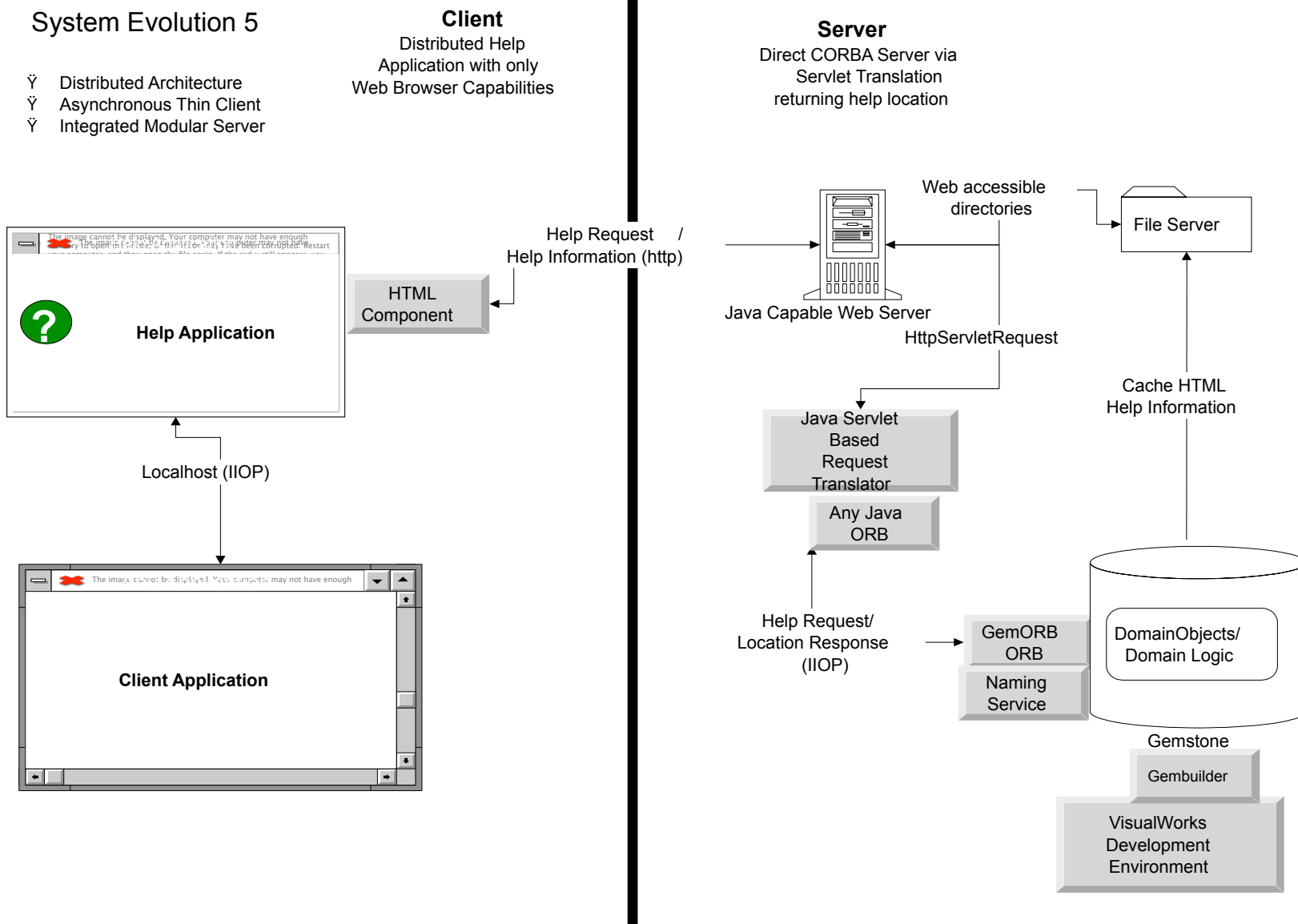
Architecture IV

- Pros:
 - Uses CORBA for Remote Messaging
 - Asynchronous Thinner Client with Stable Java VM
 - Memory Requirements Optimized
 - Help Becomes Guaranteed Client CORBA Service
 - Uses HTTP for Data Transfer (vs. IIOP)
- Cons:
 - Loss of Robust VisualWorks/DST Engine for Debugging
 - Content Generation Logic Still in Client

Architecture V

System Evolution 5

- Distributed Architecture
- Asynchronous Thin Client
- Integrated Modular Server



Architecture V

- **Pros:**
 - Uses CORBA for IPC
 - Asynchronous Thin Client
 - Java Servlets Filter http Requests Into CORBA Requests, Allowing for Thinnest Client Possible
 - Content Generation Logic Partitioned to Server
 - Memory Requirements Optimized
 - Uses HTTP for Data Transfer (instead of IIOP)
- **Cons:**
 - Loss of Robust VisualWorks/DST Engine for Debugging

Why Server Smalltalk?

- Productivity Benefits (Debugging, Development)
- Open (dll/c, sockets, RPC, CORBA)
- Robust
- Mature Class Libraries
- Mature ORBs and COS Services
- Many Mission-Critical Deployments
- Ease of Designing New Algorithms (Caching, Searching)

Summary/Conclusions

- Clients (Java/Browsers seem to have won the client wars)
- Server Smalltalk is Viable
 - Maturity
 - Proven
 - Premier Development Environment
 - Extensible
- Smalltalk ORBs (DST, GemORB) Inter-operate Well With Other (Java, C++) ORBs

Summary/Conclusions (cont.)

- **Servlets Enabled:**
 - The right protocol at the right time
 - IIOP for stable connections (e.g. IPC)
 - HTTP for less stable connections (e.g. Internet)
 - Thinner Client
 - HTTP libraries are part of Java runtime
 - Imply server side processing
- **CORBA Architectures**
 - Good for Command and Control
 - High-Value Messages
 - Are Not Good for Bulk Data Transfer
 - Integrate Heterogeneous Languages and Platforms