# HP e3000 as an E-Commerce Database Server

Greg Gibbons
MIS Director
See's Candies, Inc.

# Sees.com – B2C Since 1998

- See's Candies, Inc. is a manufacturer and retailer of fine boxed chocolates, well known in the West
- See's began selling online in 1998 using a solution provided by a third party
- In September of 2000, See's replaced the third party solution with an internally built system based around the HPe3000
- Since then customers have placed over 180,000 orders on the HPe3000 based sees.com

# Project Goals

- Tightly integrate with HP e3000 based Mail Order System
- Improve site reliability
- Move to scalable platform
- Bring the technology in house
- Allow international shipments
- Improve shipping options

# Goal: HP e3000 Integration

- HP e3000 Mail Order system
  - Mature HP e3000 system (since 1979)
  - Order Processing
  - High Capacity Order Fulfillment
  - 24 x 7 Customer Service
- Corporate Product Database
- Wide Area Mirroring

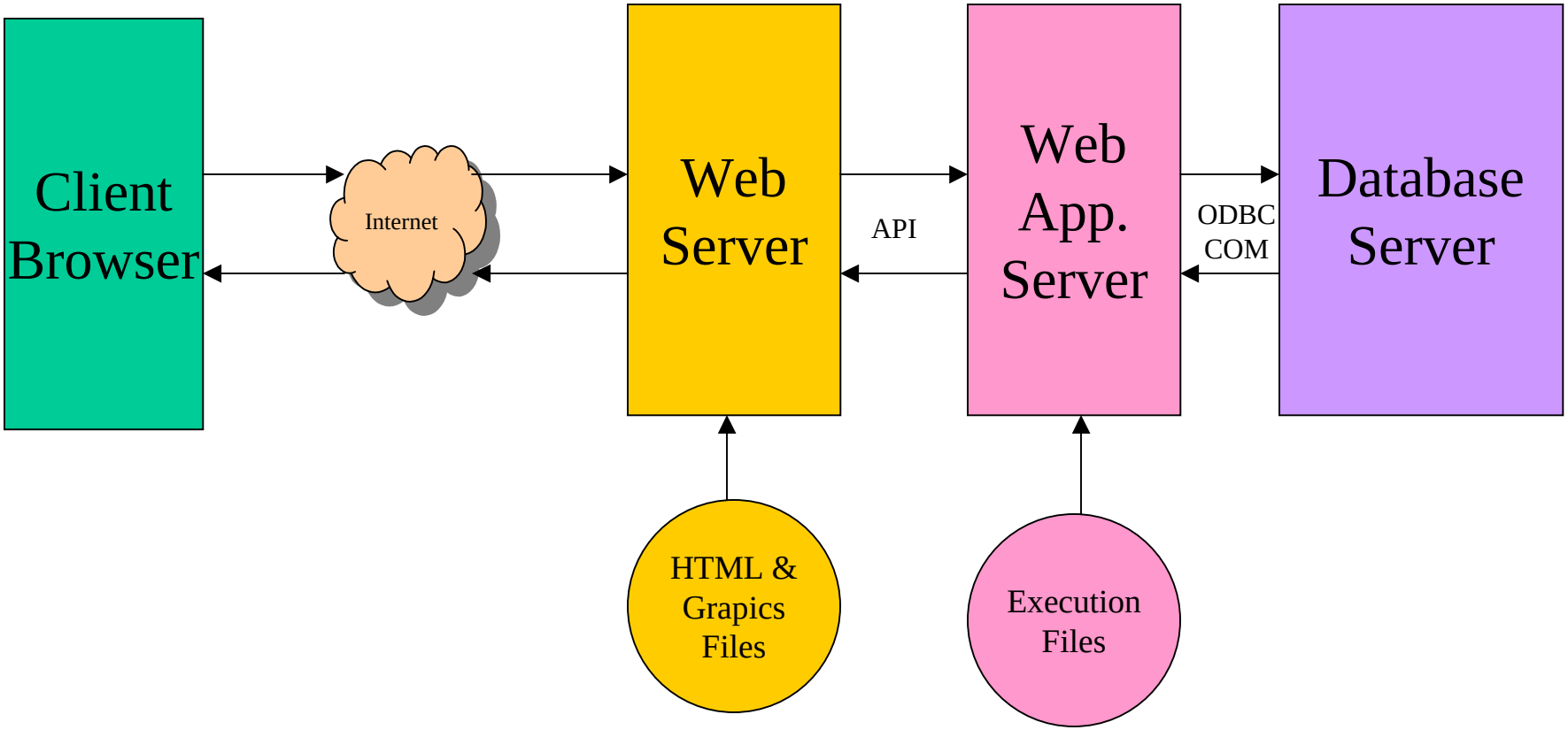
# Goal: Improve Reliability and Scalability

- Old sees.com site built on technology not suited to OLTP
- Under heavy loading
  - Orders would be lost
  - Site would slow
  - Duplicate orders created due to insufficient locking strategy
- Based on single server
  - No load sharing
  - No redundancy

# Goal: Bring Technology In House

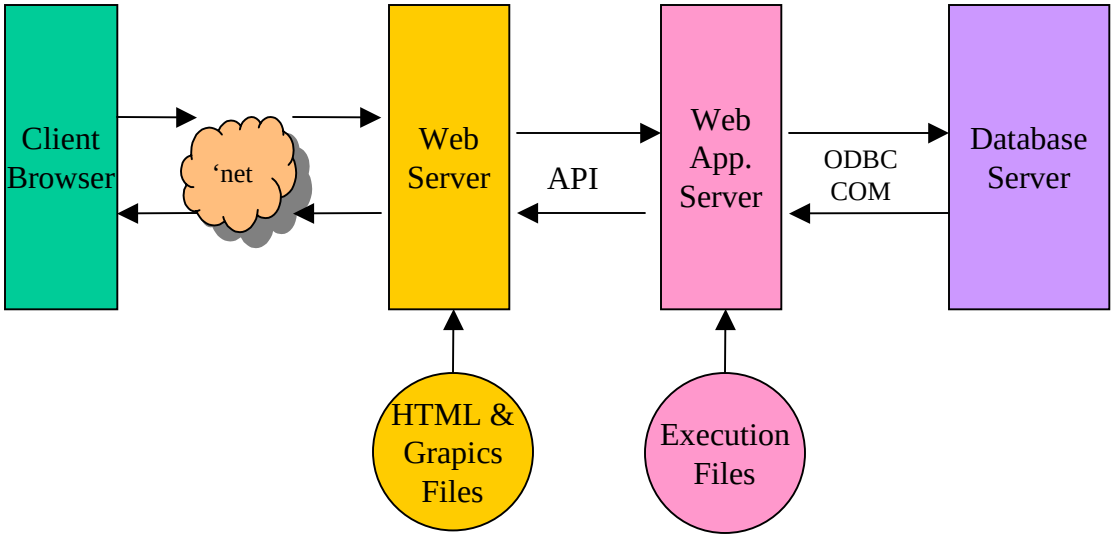
- Sees.com is mission critical and “differentiating”
- We wanted control of the software
- The Third Party we used was good, but we needed better because “nobody cares like we care”
- Technology can be applied to the See’s intranet

# N Tier Web Model



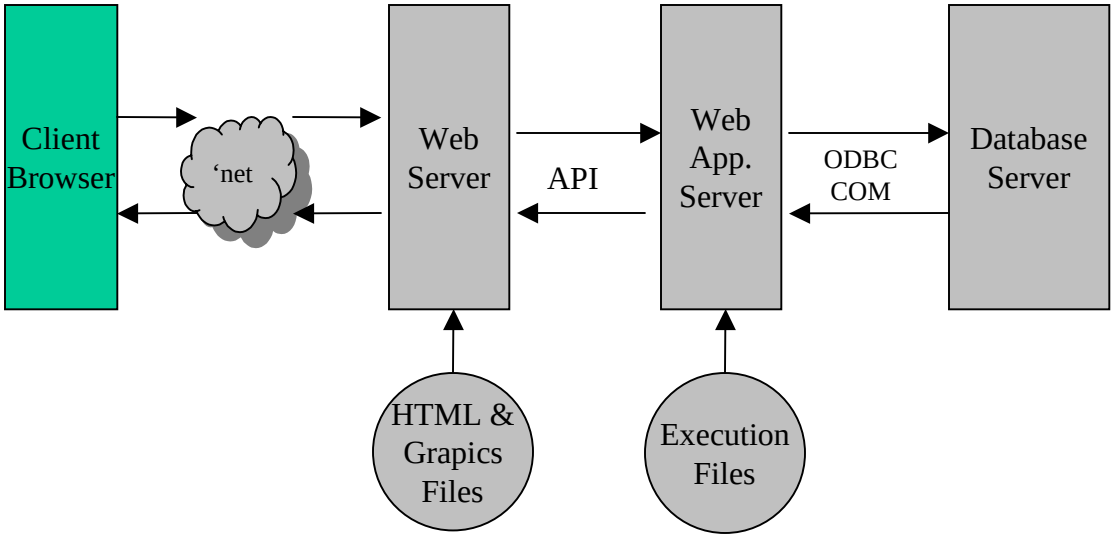
# N Tier Web Model (continued)

- Work load is distributed across tiers
- Easily scalable
- Allows each tier to be “best of breed”
- Easy to build in redundancy



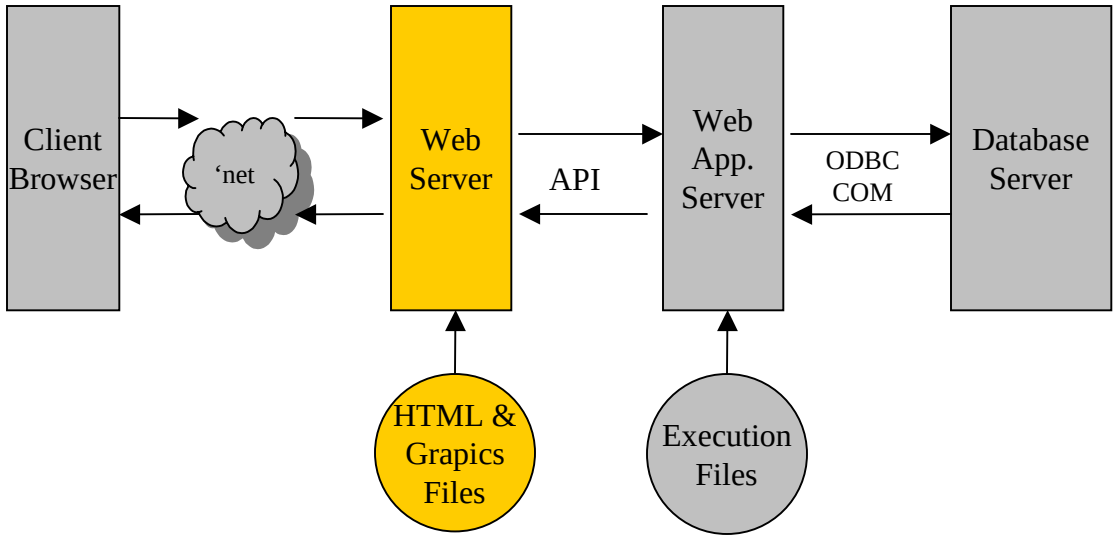
# Client Tier

- On the internet, you have little control here
- May different browsers
- Unknown browser preference and security settings
- At the mercy of AOL caching and graphics optimization
- Difficult to answer questions like “How many simultaneous users will access the system on the busiest shopping days during the Christmas season?”



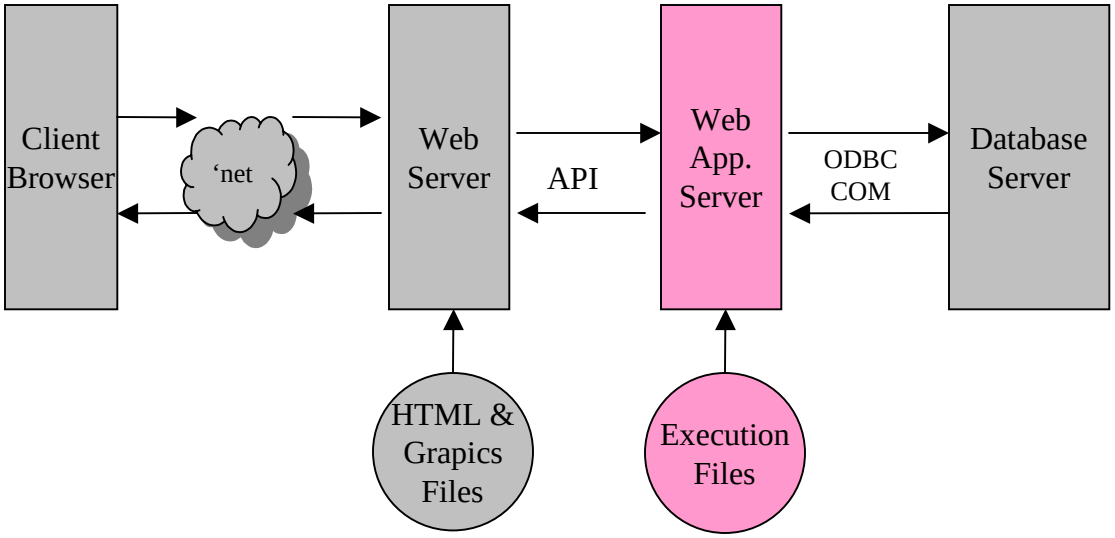
# Web Server Tier

- Serves up static HTML pages
- Passes scripts to the web application server via API for processing
- Receives rendered HTML from web application server
- Manages Secure Socket Layer (SSL)
- There are many highly tuned, cost effective solutions for this tier



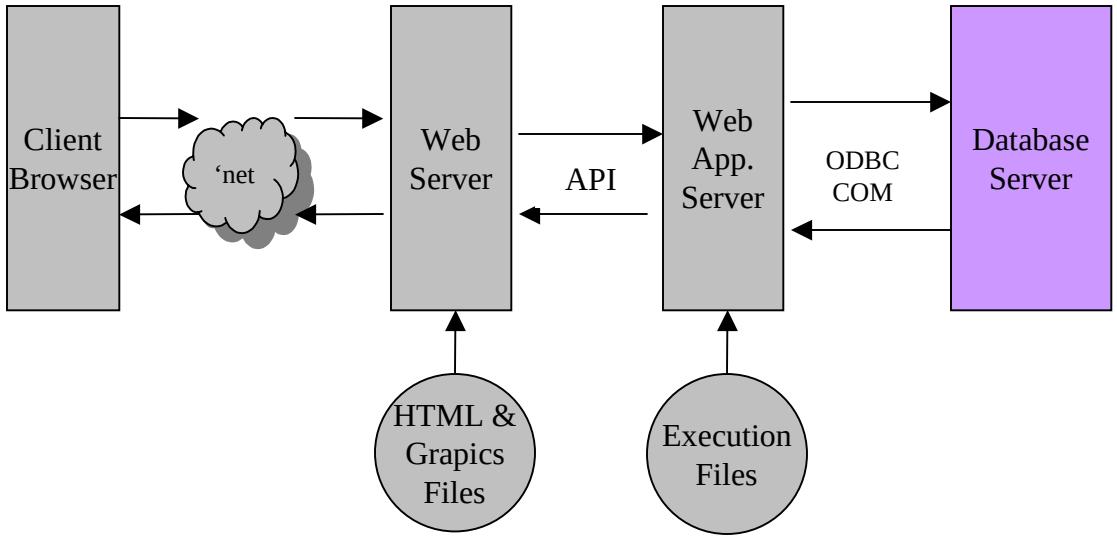
# Web Application Server Tier

- Provides an environment for rapid web development
- Offers many built in web specific features
  - Database connection pooling
  - Variety of caching options
  - Session management
- Using a web application server is a good match for medium volume sites



# Database Server Tier

- This is the most critical tier
- This tier is difficult to run in parallel so uptime and stability are critical factors



# Web Application Server research

- We knew that we wanted the HP e3000 as our database server
- We expected to use either Apache/Unix or IIS/NT for the Web Server
- We focused our attention on Web Application Server Platforms and found
  - Most required Java
  - Many were very expensive
  - Many were platform dependent

# Hmmm…Java

- Has a significant learning curve
- Being new to the web, we knew we must also learn:
  - HTML and JavaScript
  - Web Server Administration
  - Web Application Server Development and Administration
  - Middleware ODBC and Objects
  - Browser Idiosyncrasies
  - Internet structure (Domain registration, DNS, co-location, etc.)
  - Load Balancing and Security
- So, to manage the amount of skills we needed to acquire, we chose to focus on application servers that didn't require Java.

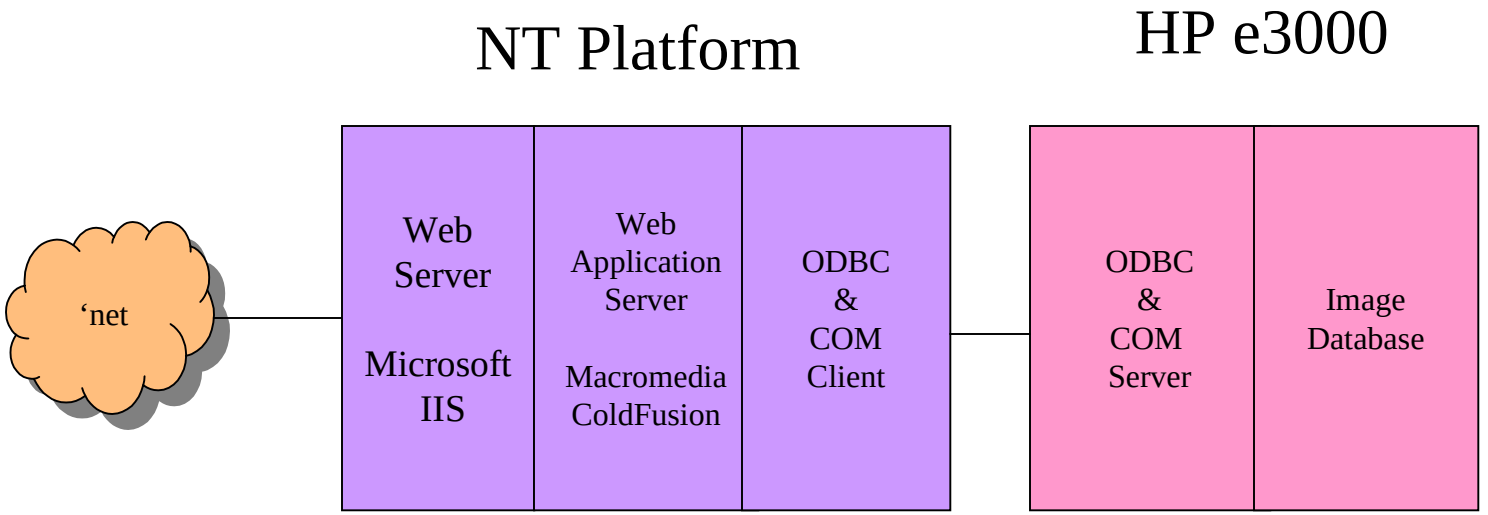
# Web Application Server Cost

- Some were as much as $25,000 per CPU (two servers with two processors each would be $100,000)
- Some required that you pay by simultaneous users, expensive and difficult to predict
- Some were $2000 per server

# Platform Independence

- Many software packages work on multiple platforms
- Experience has taught me that it is best to find out the “native” platform for a given piece of software and use that

# Initial Selection of Technologies



# Proof of Concept . . . Is this going to work?

- We set up a test bed with a dedicated NT box and an HP 3000 918RX
- We tested two ODBC Clients from different vendors
- We focused on db function and performance
- We ran long, stressful tests where the HP 3000 was running at 100% CPU until something broke
- In the end we had a combination of technologies that were fast enough and solid enough to use for ecommerce

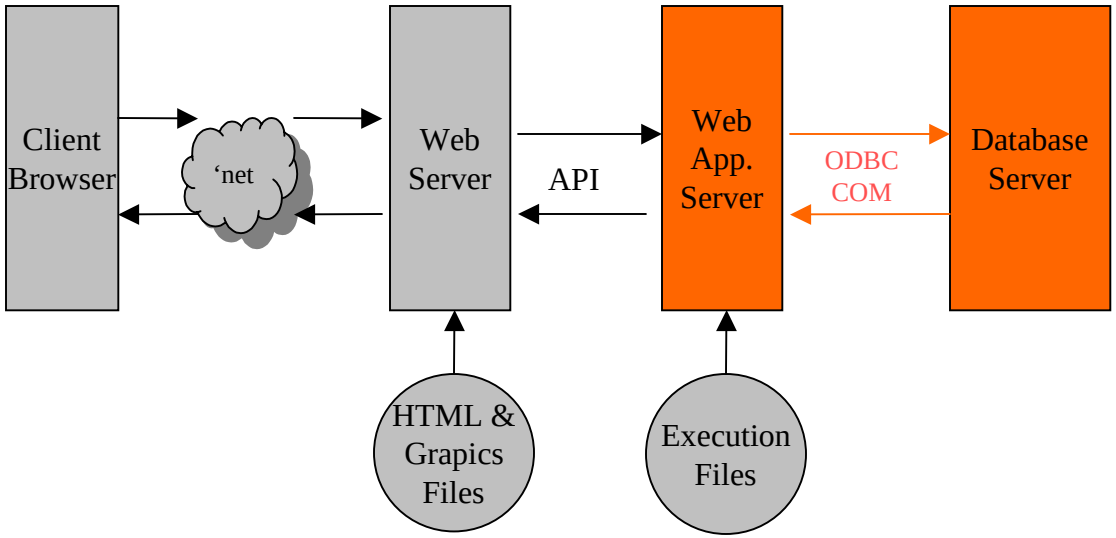



# DB Server Notes

- Remember that client/server db transactions are more expensive than traditional computing which occurs on one backplane.
- Protect the server from avoidable db accesses
- Sobering thoughts:
  - Our Mail Order system has a 400 user max. (predictable)
  - *Millions* can hit your internet application (difficult to predict)
  - When designing ask, “Can I avoid going to the db for this?”
- Place reusable data into application wide shared memory resident structures
- Rebuild these structures as needed to “pick up” changes
- Don’t scrimp on the db server you buy

# Pinch Point

- The client/server interface (middleware) between the Web Application Server and the db Server is the most likely place for a bottleneck to occur



# Pinch Point (cont.)

- c/s db drivers *on all platforms* are notorious for:
  - Bugs
  - Memory leaks
  - Thread safety issues (the hated “locked thread”)
- Select and test your drivers well
- db opens are extremely expensive in both CPU cycles and latency
- db connection “pooling” is the answer
  - Some Web Application Servers offer this
  - Also provided by some middleware

# COM Notes

- ODBC or Native db access breaks down when
  - You have db intensive transactions
  - Transactions where db locking is required
  - You want to encapsulate and/or reuse complex logic
- We use COM objects to
  - Encapsulate our complex shipping algorithms
  - Manage the end of order process which require db record locking
- Packaged COM object library prevented us from having to write our own
- COM objects must be written with thread safety in mind

# Build It

- We mocked up the navigation and web pages for shopping
- We developed and refined detailed specifications for the catalog and checkout process
- We worked with our graphic artists to “make over” the look of our site
- We selected a co-location facility
- We put together the web infrastructure

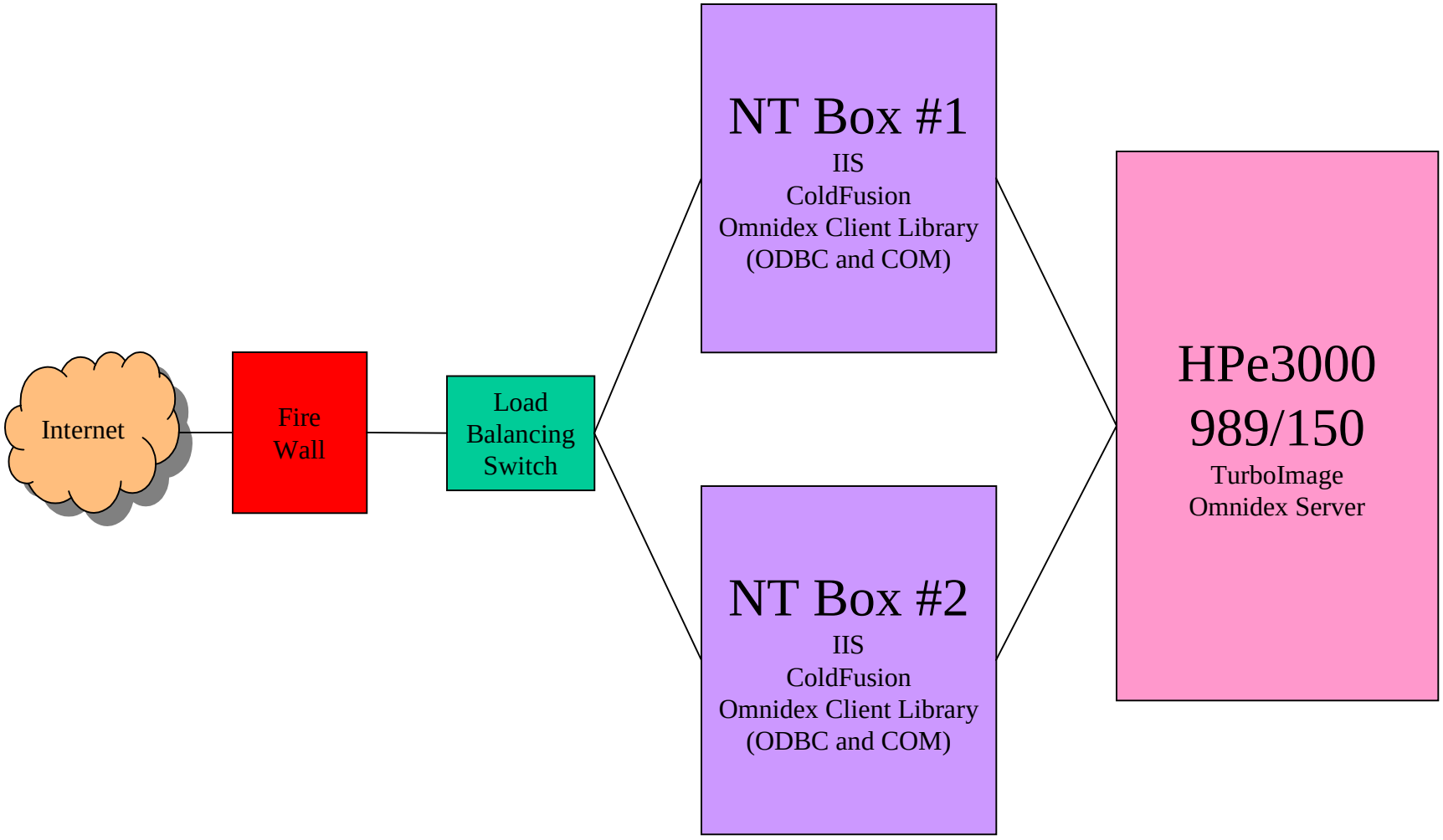
# Performance Testing

- Performance Testing/Tuning Gig
  - Most Web Application Server vendors offer this service
  - Stresses your application within an inch of its life
  - Test your *production* environment
  - Uncovers bottlenecks
  - Reveals things that fail only under stress
  - Tune the Web Application Server-to-db server relationship
  - Took a week and a bunch of money
  - Worth every penny
  - In the test, we clocked our site at 3500 orders/hour

# What did we end up with?

- HPe3000 as the data base server
  - 989/150, 2 GB RAM, Hardware RAID
  - Quest Inc, Netbase shadowing
- Middleware from Disc, Inc.
  - ODBC driver
  - COM object library
- ColdFusion web application server from Allaire (now Macromedia)
- IIS web server from Microsoft
- Load balancing switch from Alteon

# Production Configuration



# Learning Quickly

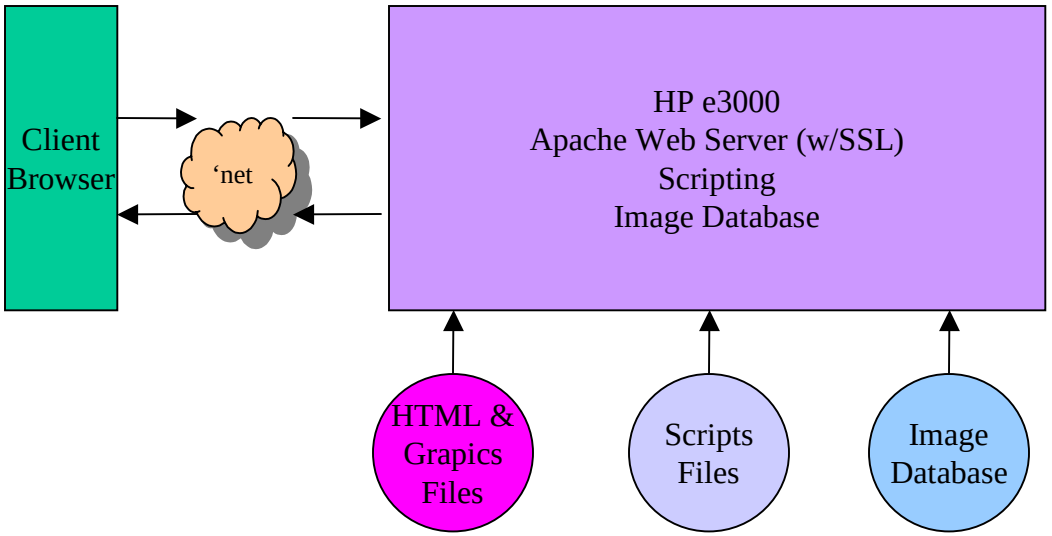
- Classes, books, and listserves
- Solid test environment for “playing”
- Web Application Server Consultant visit at start of coding (to point out “best practices”)
- Consultants again for code review (so what stupid things did we do?)

# For See's, Christmas is everything!

- Approximately 1/3 of our volume falls into December
- There is an incredible peak in web business about 12 days before Christmas
- We had very few problems
- We handled the volume well
- Since going live in September we have processed over 180,000 web orders on www.sees.com

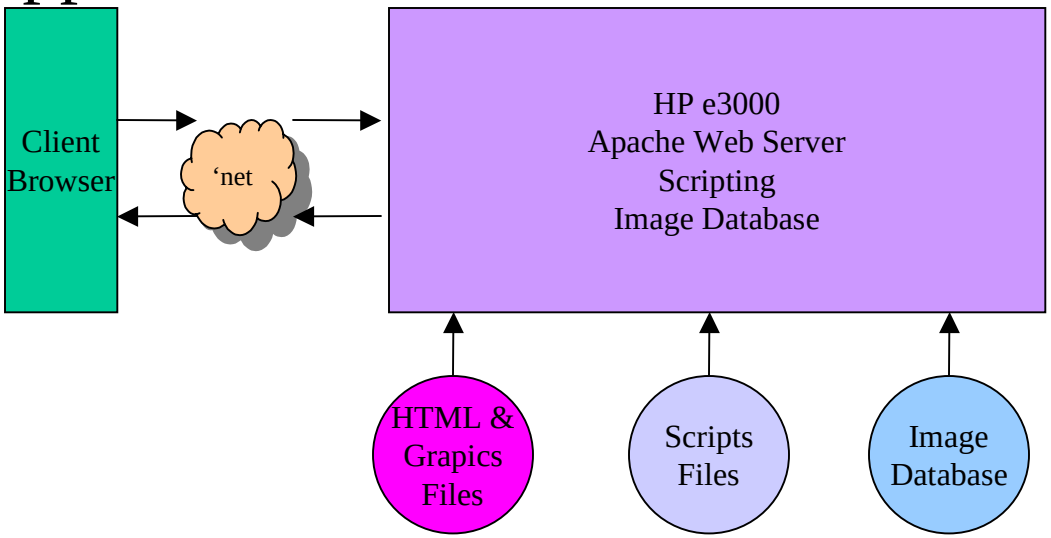
# Why not an all HPe3000 solution?

- It was 1999 and Apache on the 3k couldn't go secure (SSL)
- We wanted the productivity that the Web Application Server and it's IDE offered



# Why not an all HPe3000 solution? (cont.)

- There are highly tuned web servers on other platforms at a very attractive price point
- The web server and app. tiers will be redundant, HP e3000 reliability not required
- Tiered approach scales well



# Thank You!