

# Quality Attribute Specification

- Architects are often told:
  - “My application must be fast/secure/scale”
- Far too imprecise to be any use at all
- Quality attributes (QAs) must be made precise/measurable for a given system design, e.g.
  - *“It must be possible to scale the deployment from an initial 100 geographically dispersed user desktops to 10,000 without an increase in effort/cost for installation and configuration.”*

# Quality Attribute Specification

- QA's must be concrete
- But what about testable?
  - Test scalability by installing system on 10K desktops?
- Often careful analysis of a proposed solution is all that is possible
- “It's all talk until the code runs”

# Performance

- Many examples of poor performance in enterprise applications
- Performance requires a:
  - Metric of amount of work performed in unit time
  - Deadline that must be met
- Enterprise applications often have strict performance requirements, e.g.
  - 1000 transactions per second
  - 3 second average latency for a request

# Performance - Throughput

- Measure of the amount of work an application must perform in unit time
  - Transactions per second
  - Messages per minute
- Is required throughput:
  - Average?
  - Peak?
- Many system have low average but high peak throughput requirements

# Performance - Response Time

- measure of the latency an application exhibits in processing a request
- Usually measured in (milli)seconds
- Often an important metric for users
- Is required response time:
  - Guaranteed?
  - Average?
- E.g. 95% of responses in sub-4 seconds, and all within 10 seconds

# Performance - Deadlines

- 'something must be completed before some specified time'
  - Payroll system must complete by 2am so that electronic transfers can be sent to bank
  - Weekly accounting run must complete by 6am Monday so that figures are available to management
- Deadlines often associated with batch jobs in IT systems.

# Something to watch for ...

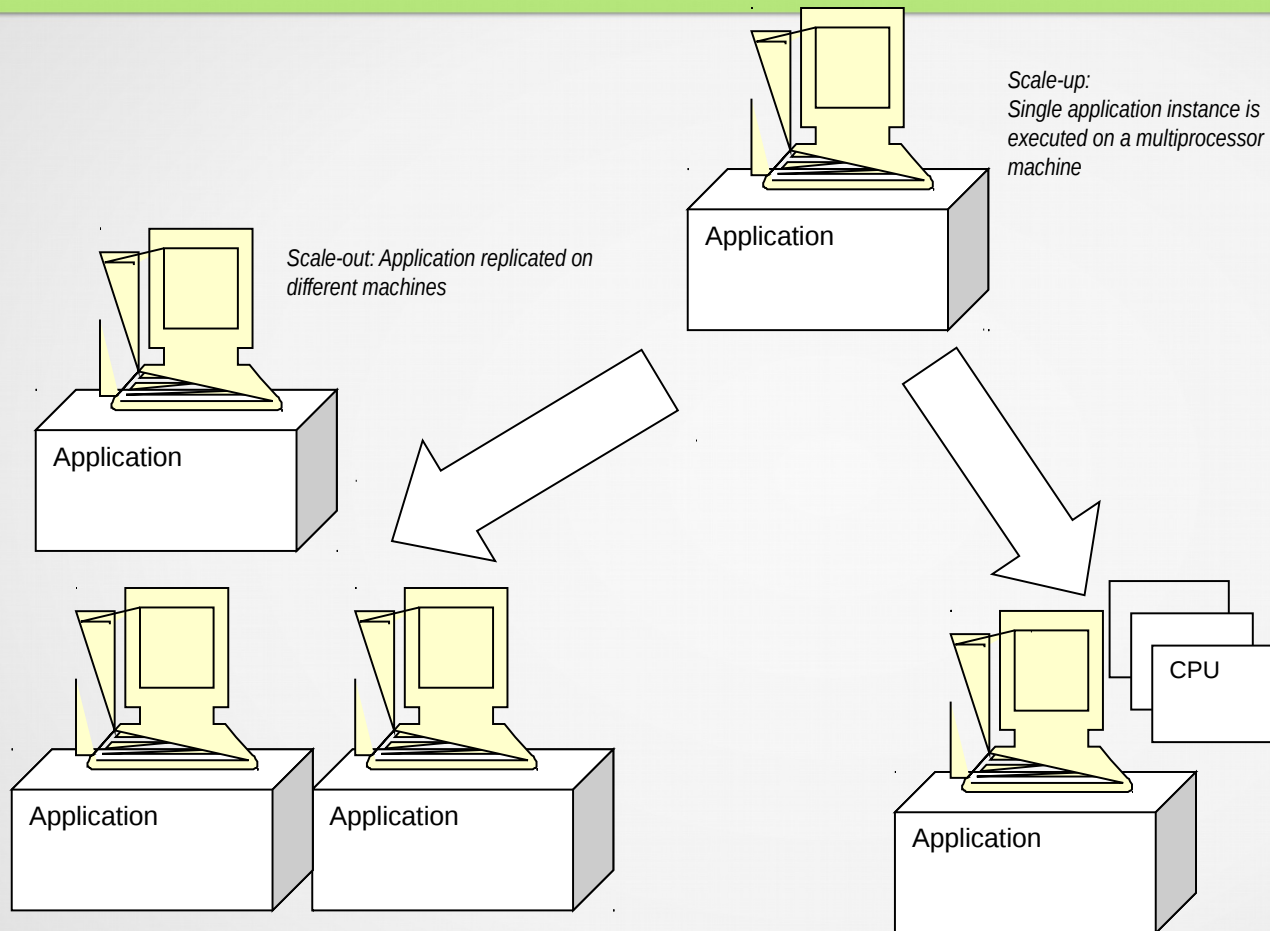
- What is a
  - Transaction?
  - Message?
  - Request?
- All are application specific measures.
- System must achieve 100 mps throughput
  - BAD!!
- System must achieve 100 mps peak throughput for *PaymentReceived* messages
  - GOOD!!!

# Scalability

- *“How well a solution to some problem will work when the size of the problem increases.”*
- 4 common scalability issues in IT systems:
  - Request load
  - Connections
  - Data size
  - Deployments



# Scalability – Add more hardware ...



# Scalability - reality

- Adding more hardware should improve performance:
  - scalability must be achieved without modifications to application architecture
- Reality as always is different!
- Applications will exhibit a decrease in throughput and a subsequent exponential increase in response time.
  - increased load causes increased contention for resources such as CPU, network and memory
  - each request consumes some additional resource (buffer space, locks, and so on) in the application, and eventually these are exhausted
  - Data almost never completely independent

# Message Brokers

- Developed specifically for Enterprise Application Integration (EAI)
- Add new layers of functionality to MOM
  - Message transformation
  - Rules engine
  - Intelligent routing
  - Adapters
- Typically (but not necessarily) built on top of a MOM layer

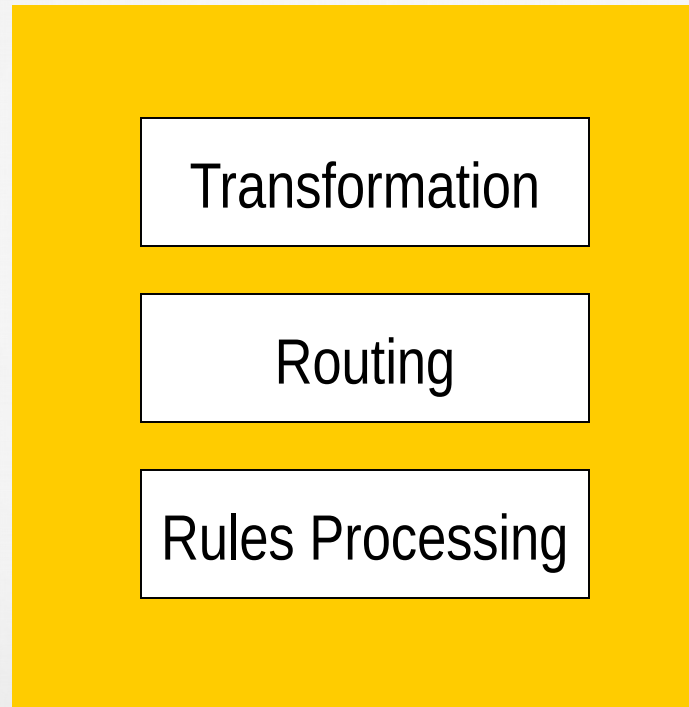
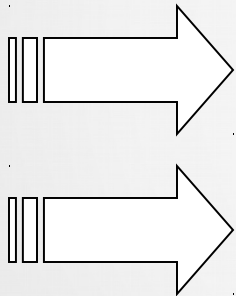
# Message Broker Features

- Message transformation – transform between different source/target formats
  - Graphical message format definition and mapping tools
  - High performance transformation engines
  - Message format repositories
- Intelligent routing
  - Route messages based on message content
- Rules Engine
  - Scripting language, built-in functions
  - Application programming environment

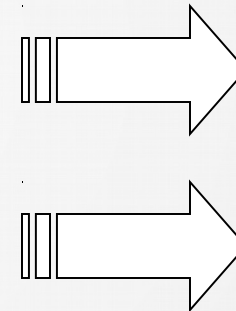
# Message Brokers

## Hub and Spoke Architecture

Input Messages



Output Messages



# Adapters

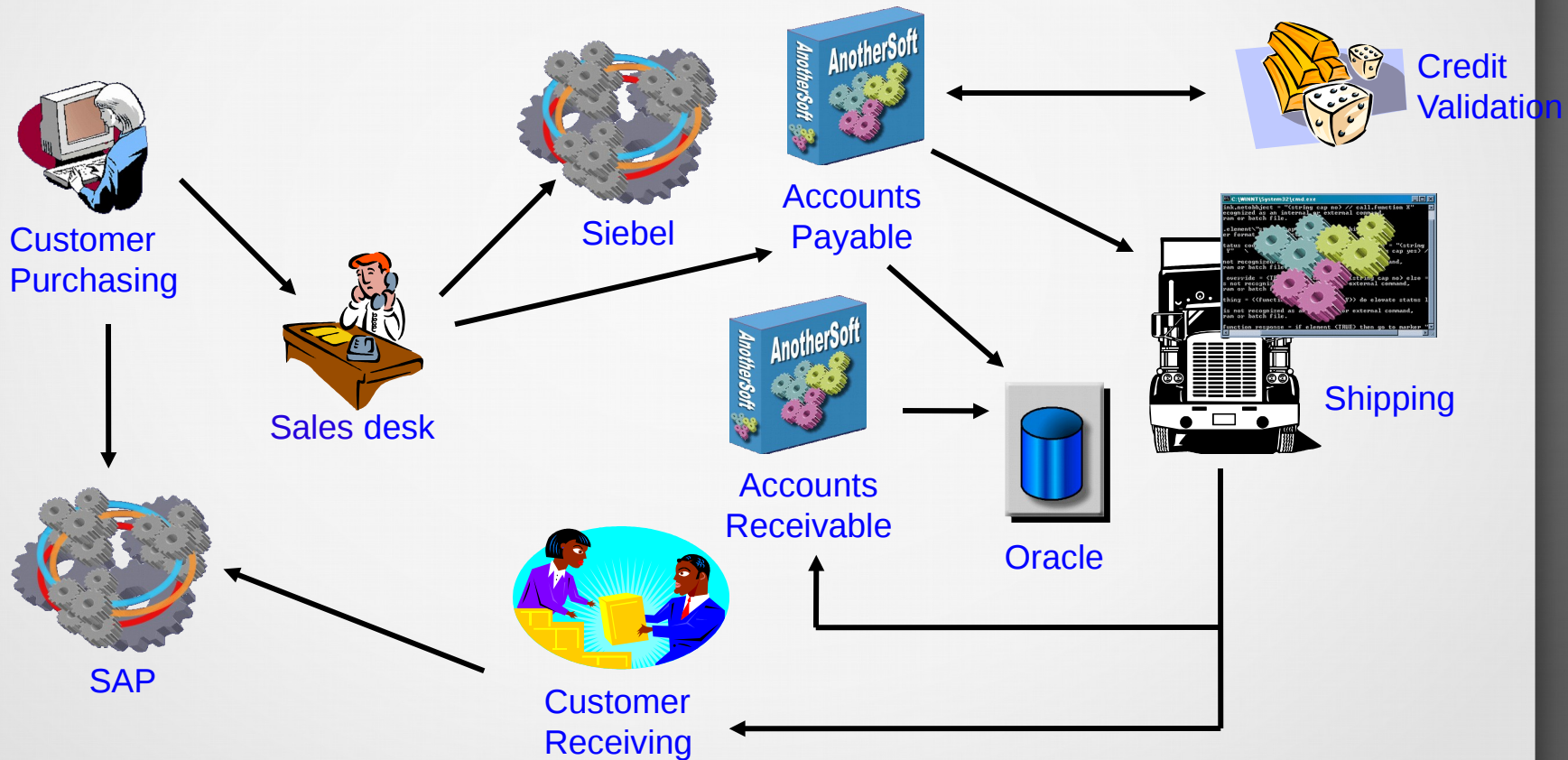
- An adapter is a component that resides between the message broker and the source/target systems
- Simplify complexity of end system interface through an abstraction layer
- Thin adapters - simple wrappers
- Thick adapters
  - Programmable
  - Abstract representation of services and meta-data
- Centralized adapters co-located with broker
- Distributed adapters execute in own process and may be located with source/target system

# Business Process Orchestration

- Commonly known as workflow
- Aim is to automate business processes which need to access data and business logic across disparate back-end applications
- Builds on EAI to ensure business processes are executed in the defined order using the required data
- Builds on middleware providing:
  - Process execution engine
  - Visual process definition tools
  - Process monitoring tools

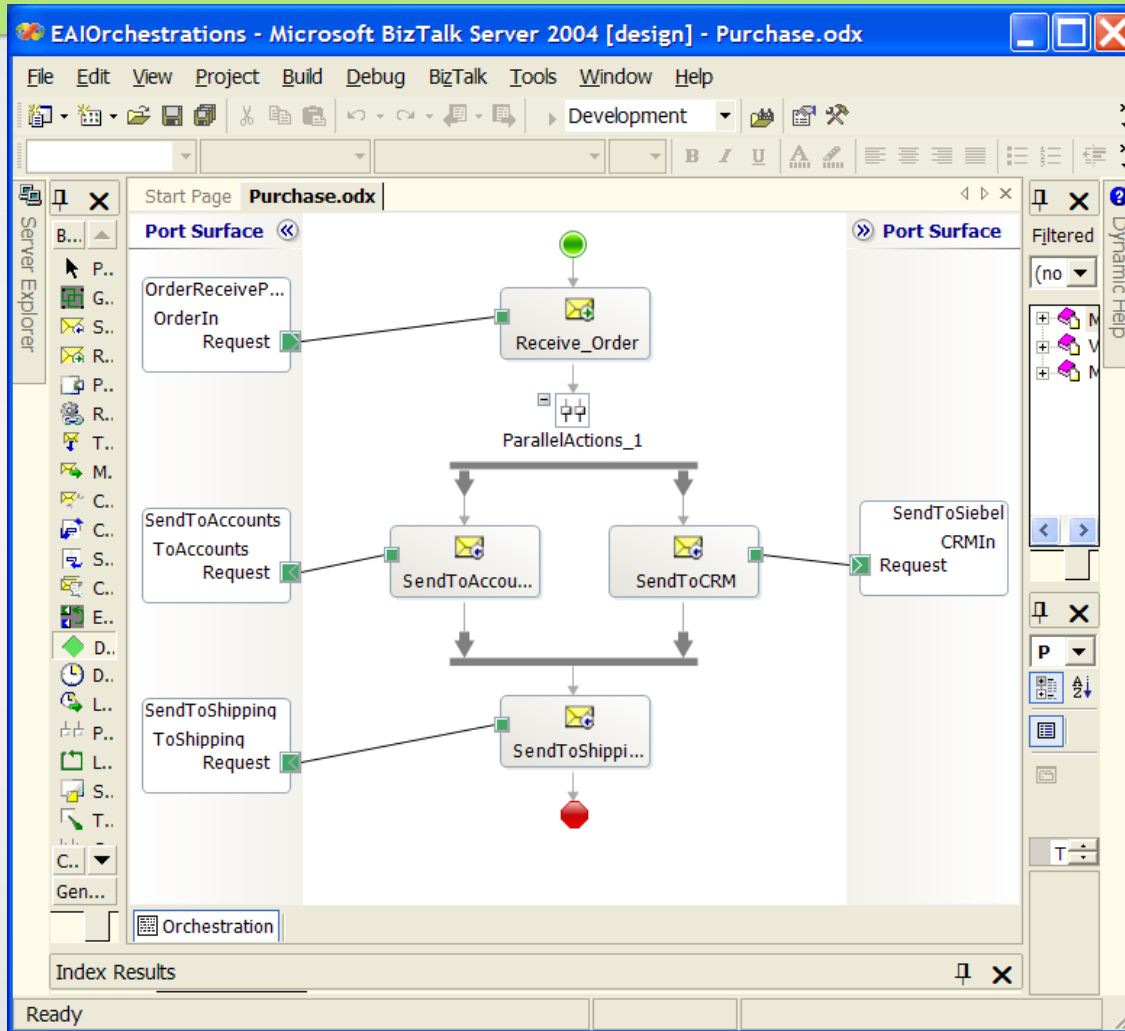
# Typical Scenario

- Business process automation

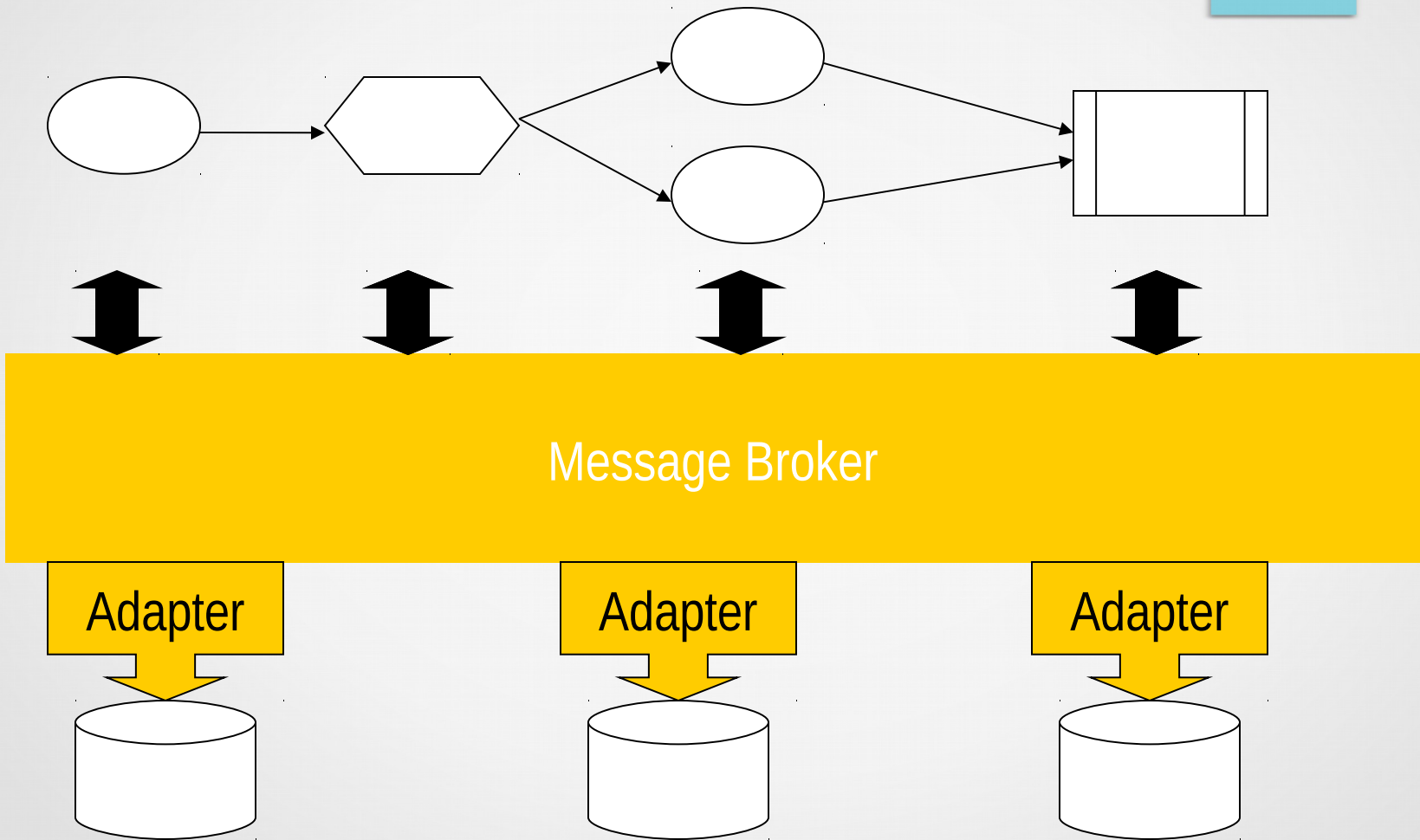




# Example - BizTalk



# BPO Architecture



# BPEL

- Web Services standard for describing workflows
- Many design and execution tools
  - Eg ActiveBPEL
- Version 2.0 is a significant improvement