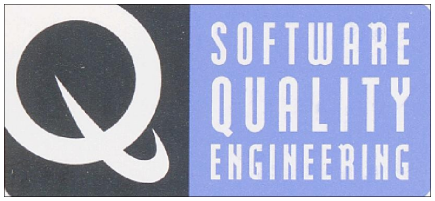
SOA Testing Challenges

First Presented for:



Webinar, 5/9/2006

Scott Barber
Chief Technologist
PerfTestPlus, Inc.

Agenda

Evolution of Application Architecture

General SOA Concepts

SOA Testing Challenges

Summary

Questions

Evolution of Software Architecture

Stand-alone

Client Server

Distributed “n-tier”

Web Services

Service Oriented

Stand Alone Architecture

Tightly coupled collections of data and function residing on a single machine.

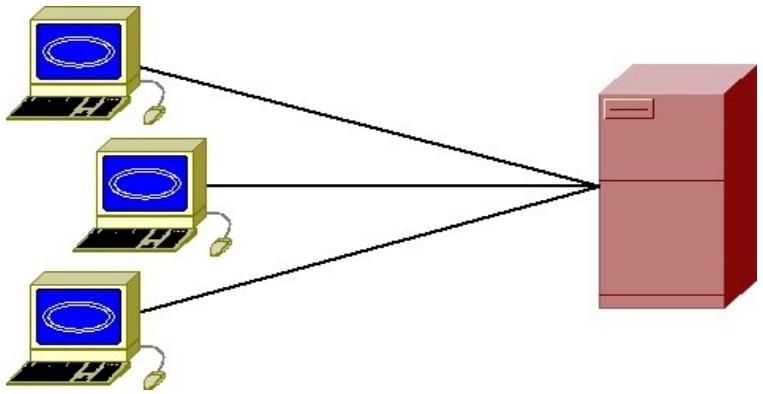
Reuse, redesign, and replacement of component parts can be very difficult.

Client Server Architecture

Similar to Stand Alone:

- **Tightly coupled collections of data and functions reside on a single machine (server).**
- **Reuse, redesign, and replacement of component parts can be very difficult.**

Unique in that the multiple users access the application from remote machines (clients).

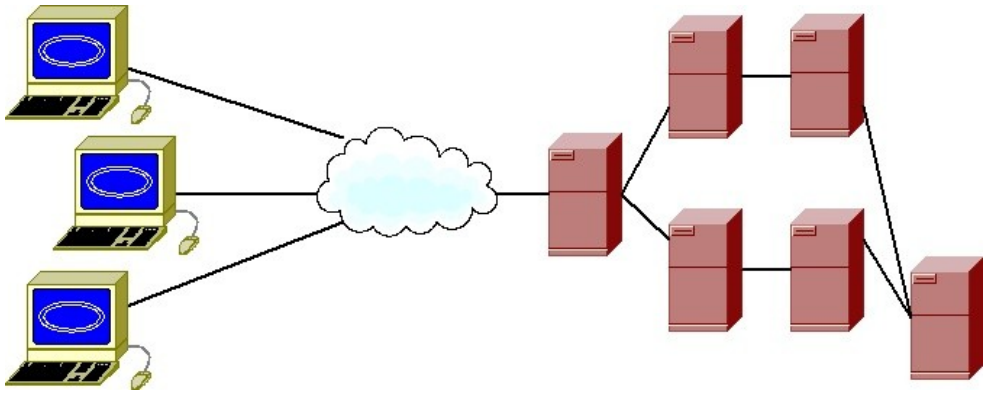


Distributed Architecture

Loosely or tightly coupled collections of data and function residing on multiple machines.

Reuse, redesign, and replacement of component parts may or may not be very difficult.

Typically serves multiple remote users.



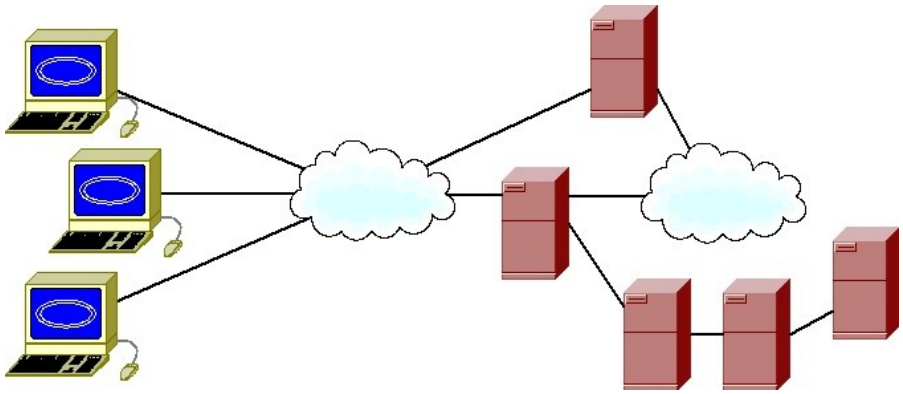
Web Services Architecture

Loosely or tightly coupled collections of data and function residing on multiple machines.

One or more collection of data and/or function built and hosted externally.

Reuse, redesign, and replacement of component parts may or may not be very difficult.

Typically serves multiple remote users.



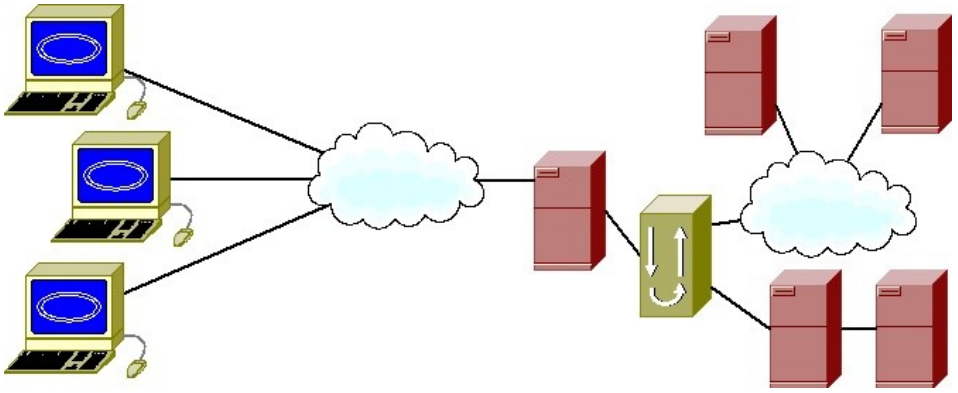
Service Oriented Architecture

Loosely coupled collections of data and function residing on multiple internal or external machines.

Often, collections communicate through a single interface.

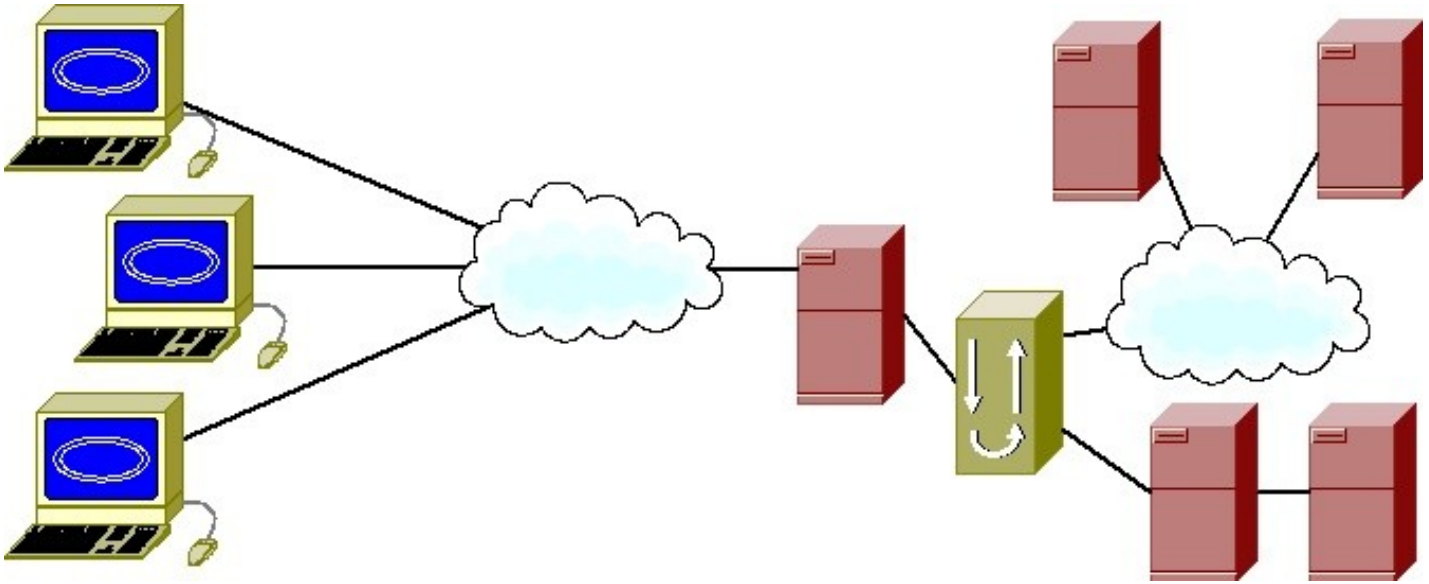
Reuse, redesign, and replacement of component parts should not be very difficult.

Typically serves multiple remote users.



The Concept Of SOA (general)

Service Oriented Architecture (SOA) is an approach to implement business systems across a loosely coupled set of technologies.



The Concept Of SOA (view 1)

SOA is “a way of designing and implementing enterprise applications that deals with the intercommunication of loosely coupled, coarse grained (business level), reusable artifacts (services) that are accessed through well-defined, platform independent, interface contracts."

- Steve Wilkes

The Concept Of SOA (view 2)

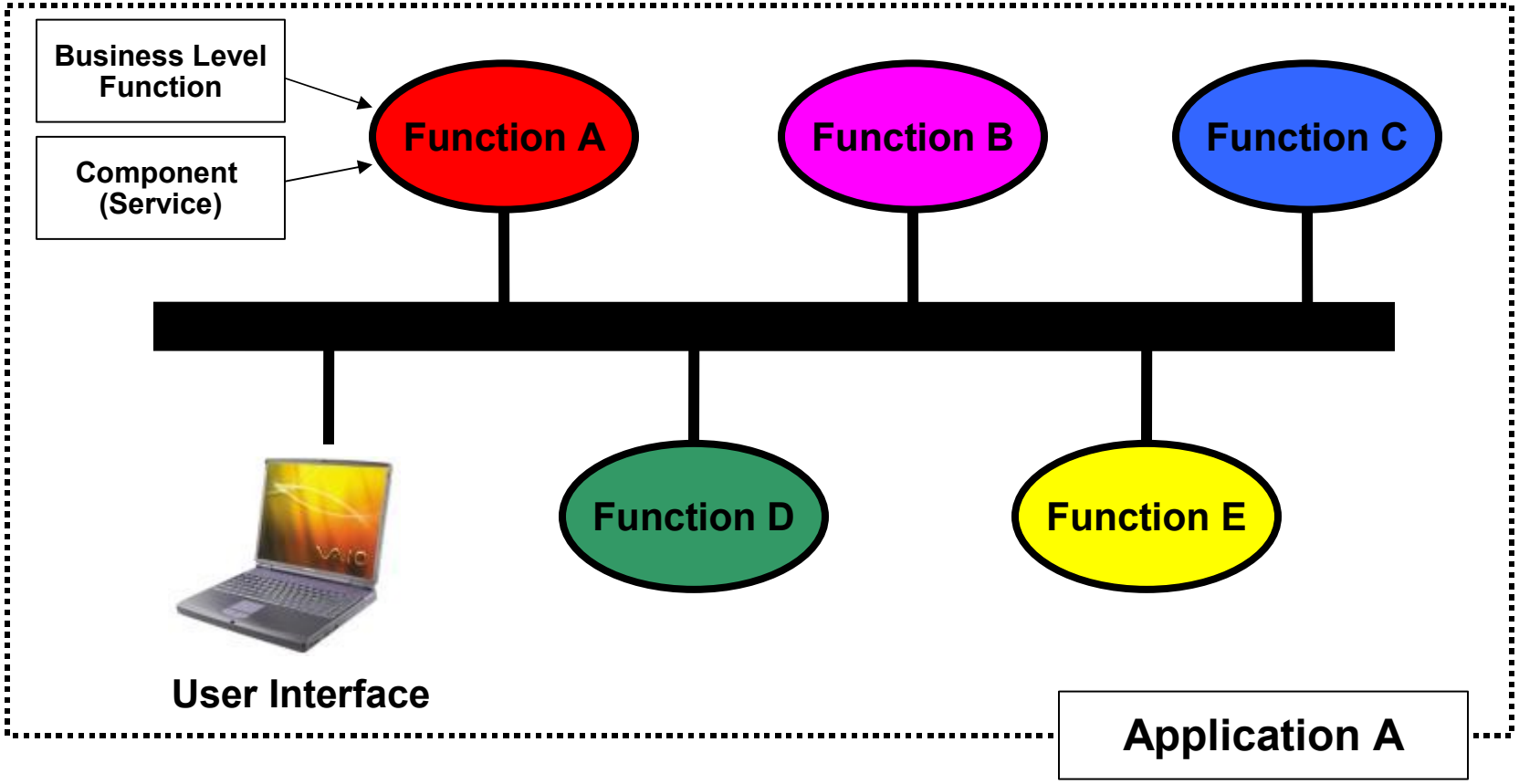
"I've heard people say the nice thing about SOA is that it separates data from process, that it combines data and process, that it uses web standards, that it's independent of web standards, that it's asynchronous, that it's synchronous, that the synchronicity doesn't matter...

... I was on the SOA panel. I played it for laughs by asking if anyone else understood what on earth SOA was. Afterwards someone made the comment that this ambiguity was also something that happened with Object Orientation. There's some truth in that, there were (and are) some divergent views on what OO means. But there's far less Object Ambiguity than the there is Service Oriented Ambiguity…"

- Martin Fowler

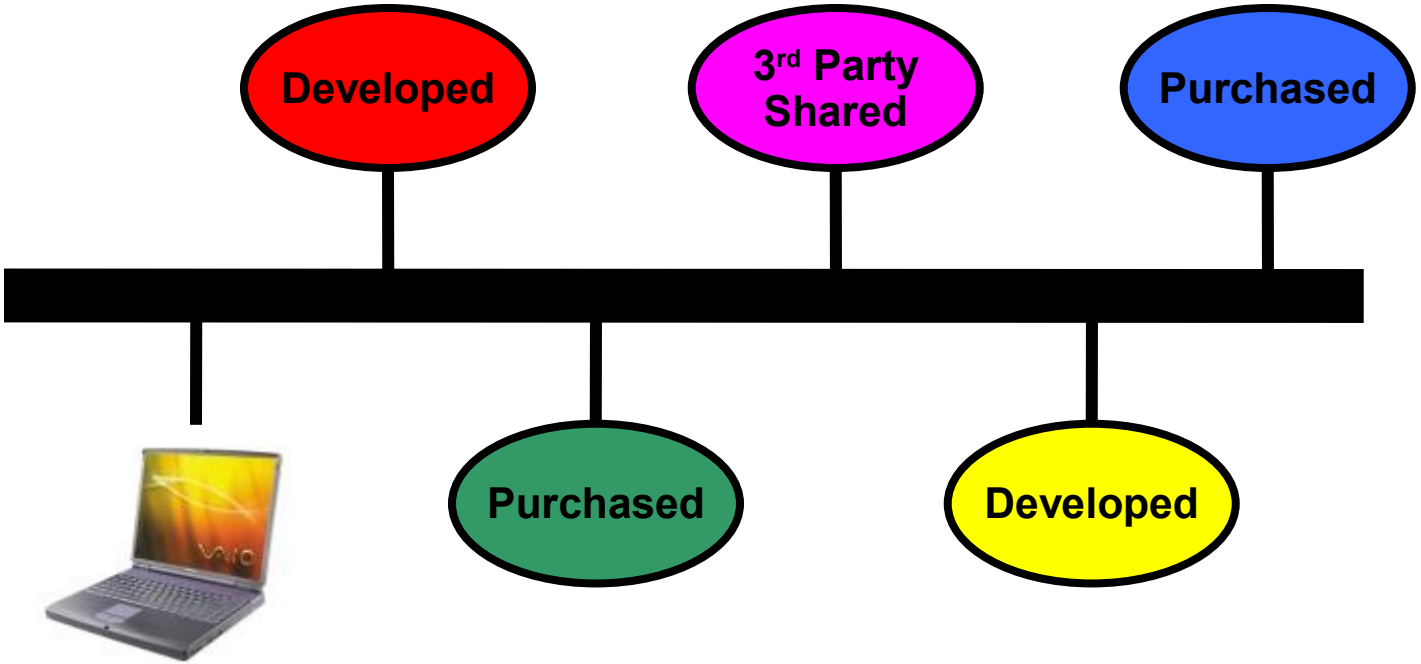
SOA Application Design

This part, everyone seems to agree on.



The Concept of SOA

In fact, these services can from many sources and be distributed over heterogeneous environments.



Innovations

The ambiguity Martin Fowler jokes about is the myriad of available choices in hardware, software, and process technologies.











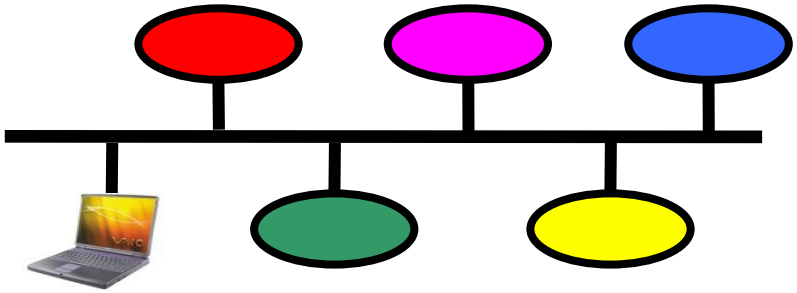


Service Oriented Architecture

Testing Challenges

SOA Testing Challenges

Since SOA applications are composed of loosely coupled, business-level services, distributed over a network, we must test the application...



end-to-end,

service-by-service

and interface-by-interface.

SOA Testing Challenges

We need to trust SOA services in the following areas, but we may not control the testing.

Functionality

- **API** (Application Programming Interface)
- **Interoperability/Integration**

Publish, Find and Bind

Security

Performance

Functionality Testing Challenges

The overall functionality of SOA applications should be easier to test.

That is only if we, or someone we trust, thoroughly test the application's components (services) before we assemble them to create the application.

Starting with lower-defect components typically means a smoother testing process BUT …

Functionality Testing Challenges

SOA applications generally have an increased number of:

- APIs (one for each service)
- Communication paths between those services

There is now an increased level of integration and interoperability testing that must be performed.

Who is responsible for this testing? The service provider? The service requester? Both? Someone else?

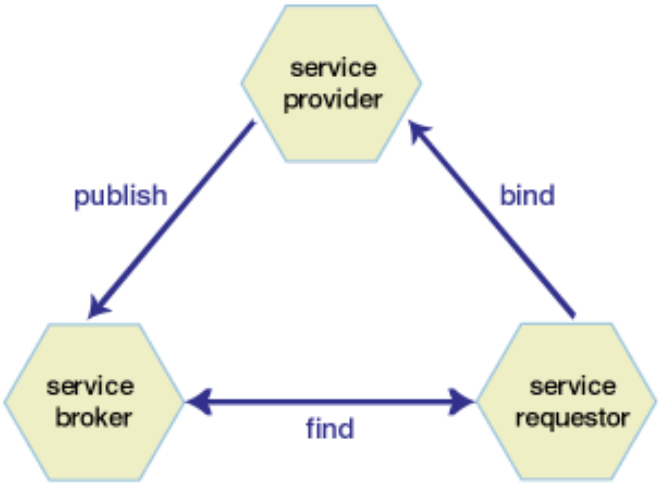
How do we establish trust?

Publish, Find & Bind Testing Challenges

Service providers must “advertise” their existence to brokers (automated catalogues).

Service requesters (SOA applications) must find providers and connect (bind) to them to use their services.

Service brokers must accept registrations and then provide information through search functions.



Publish, Find & Bind Testing Challenges

You may not have tested anything like this before... unless you've tested 3rd party credit card processing for a web application.

Are the services your organization is providing able to properly register themselves?

Can your SOA application find and properly bind with services?

Whose responsibility is it to test? The provider? The requester? Both? Someone else?

How do we establish trust?

Security Testing Challenges

An SOA application is a collection of independent services, collaborating to provide valuable functionality.

"Valuable" often suggests the need for security, for the authentication of users before access.

Consider an application in which each service requires a different authentication approach and enforces different security policies – it's a design, development, and testing nightmare.

Security Testing Challenges

Your organization needs a centralized SOA security management approach.

Various strategies include:

- Ignore the problem
- Hide within a private network
- Whip up a home-brewed solution
- Buy this functionality from an experienced vendor

Whose responsibility is it to test? The provider? The requester? Both? Someone else?

How do we establish trust?

Performance Testing Challenges

All this loosely-coupled, platform-independent stuff is not free.

Major performance problems are often due to:

- Layer upon layer; abstraction upon abstraction
- Small services with large overhead
- Large services that are under supported by hardware
- Services distributed across a network with its associated latency

One could also say...

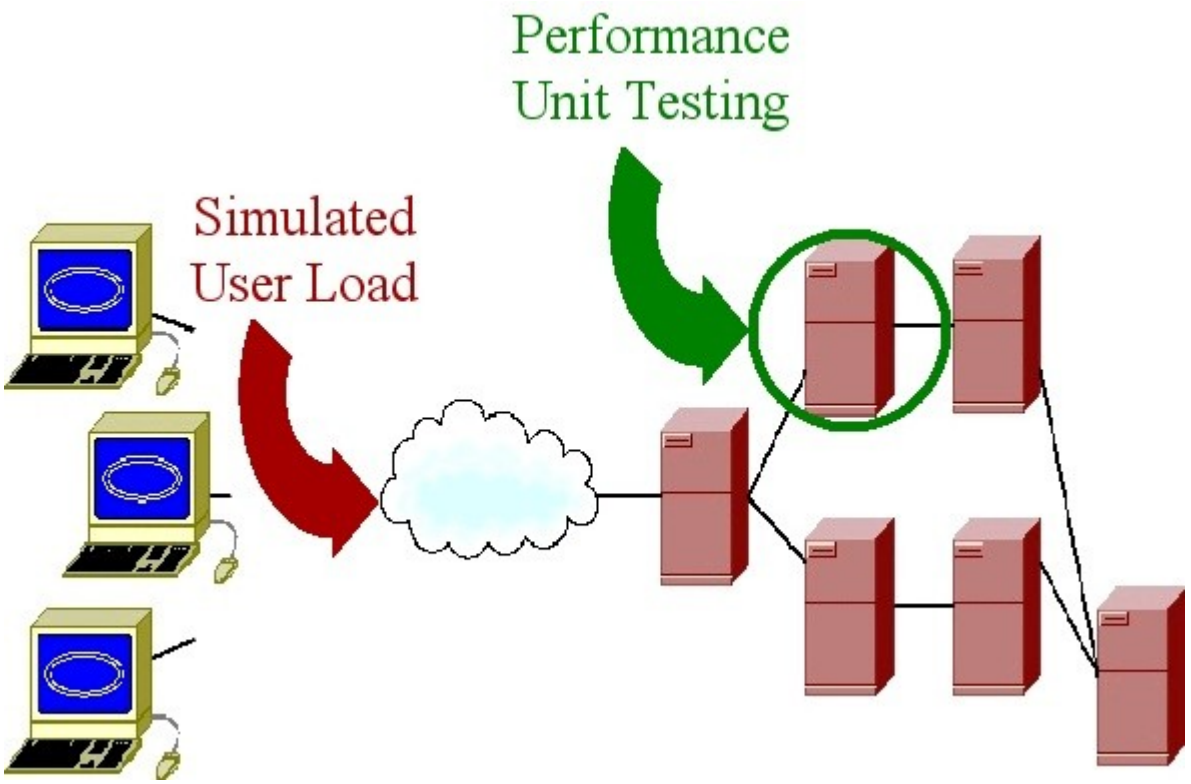
"Performance is not synergistic."

Performance Testing Challenges

The application will need to be performance tested...

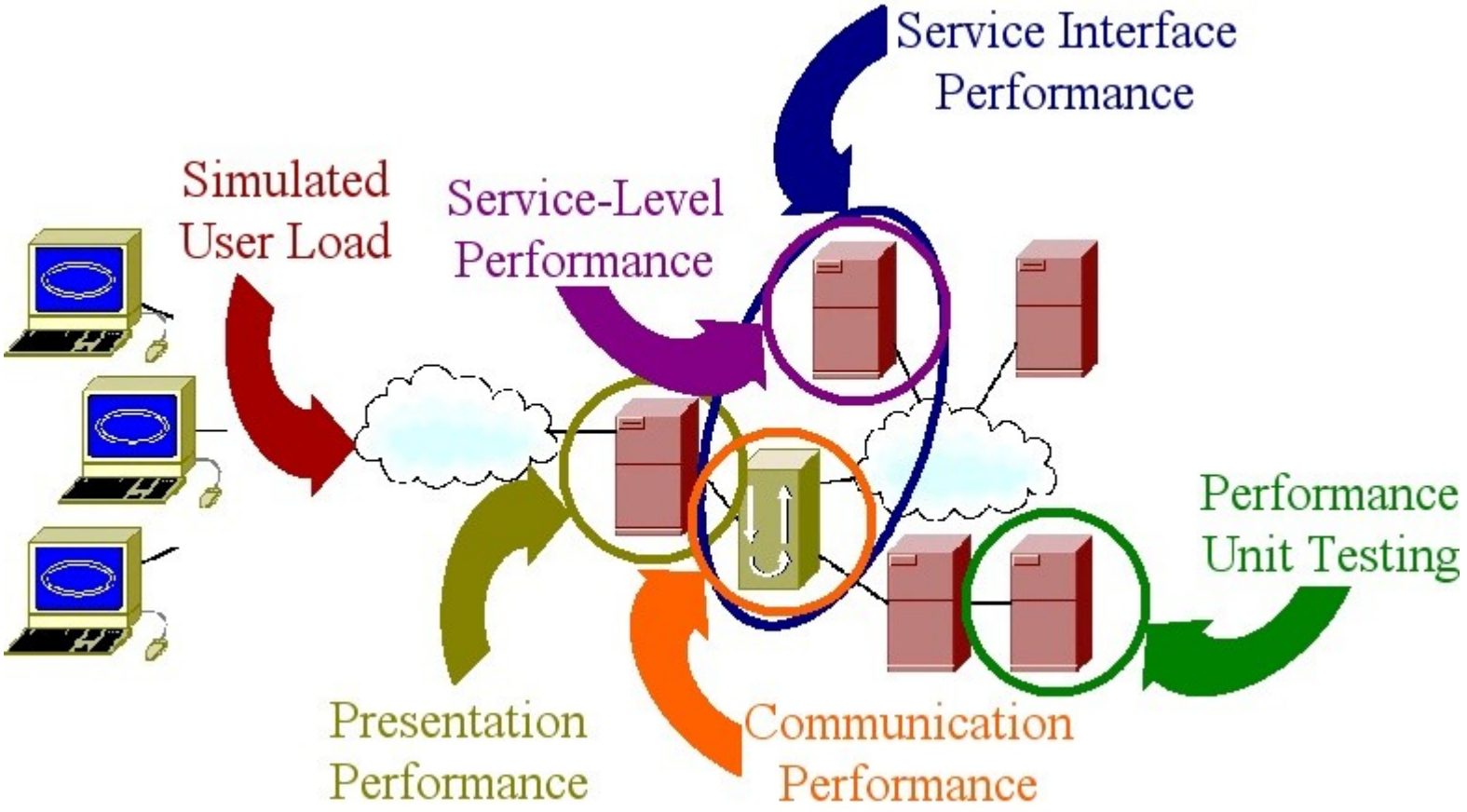
- End-to-end, from the user's perspective (requester)
- At the unit level during development (provider)
- At the service level (likely both requester and provider)
- To validate each interface (likely both requester and provider)
- To ensure correct functionality under load

Performance Testing Challenges



What we've become accustomed to.

Performance Testing Challenges



What we should have been doing all along...
and with SOA have little choice.

SOA Testing Challenges Overall

What is your degree of trust in… ?

Functionality

- **APIs**
- **Integration/Interoperability**

Publish, Find, and Bind

Security

Performance

Or Are You

SOA Testing Challenges Overall

- Whose responsibility is it to test services?
  - The provider?
  - The requester?
  - Both?
  - Someone else?
- How well did your vendor test?
  - How do you know?
- How do we establish trust in a service?
- How do we demonstrate that our SOA application is worthy of the trust of our users?

Summary Of Testing Challenges

Today we've discussed the promises and testing challenges in SOA applications:

Functionality

- APIs
- Integration/Interoperability

Publish, Find, and Bind

Security

Performance

Questions

Contact Info

Scott Barber

Chief Technologist

PerfTestPlus, Inc

E-mail:
sbarber@perftestplus.com

Web Site:
www.PerfTestPlus.com