Some Performance and Security Findings Relative to a SOA Ground Implementation

March 28, 2007

John Hohwald

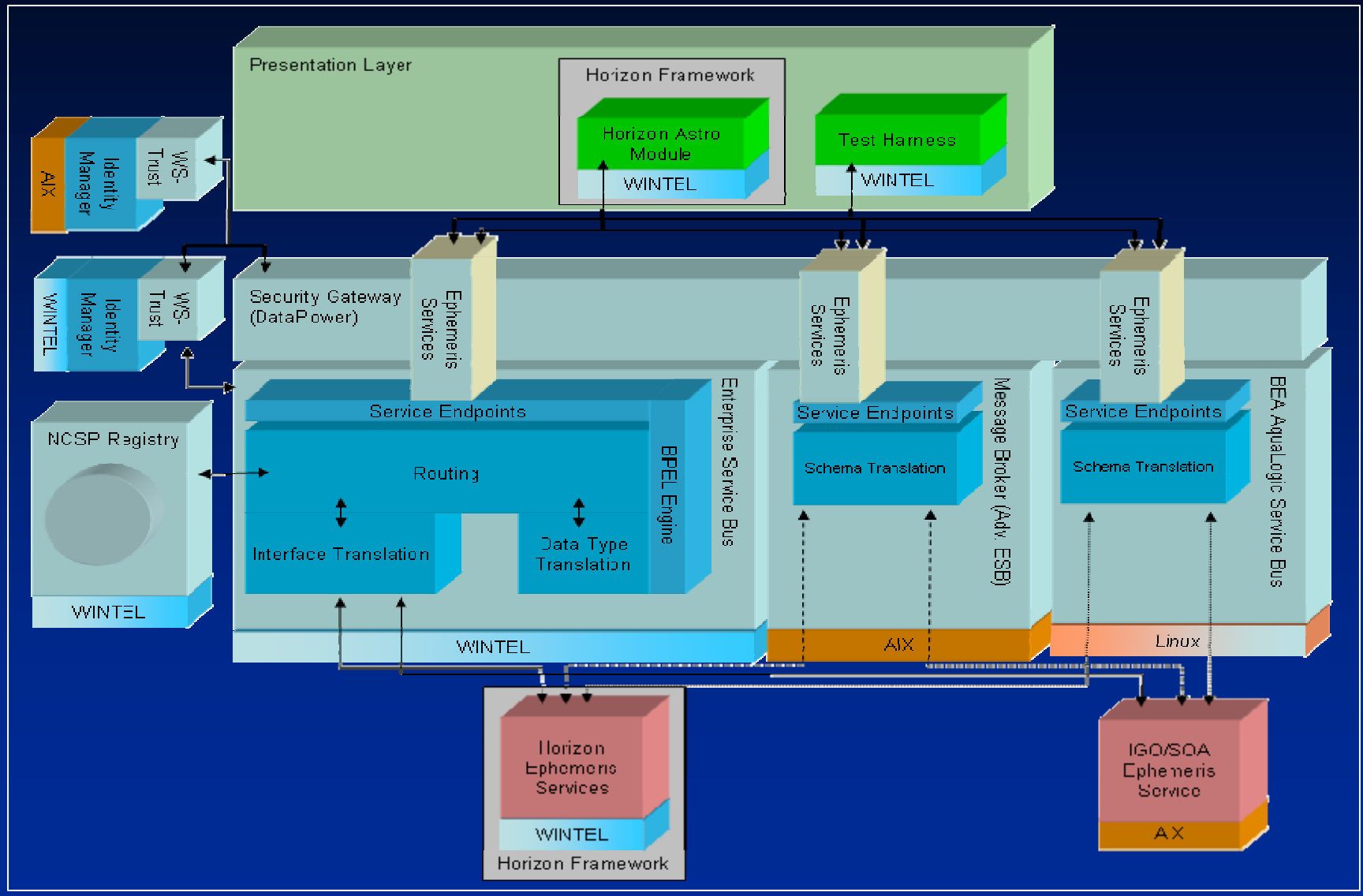
Ground SOA Implementation Issues

- SOA Benchmarking
  - Benchmarked a variety of vendors
    - IBM Websphere Process Server 6.0.1; IBM Advanced Websphere (Message Broker) 6.0.0 (on AIX 5.3)
    - DataPower XS40 (HW appliance: XML Accelerator and SAML 2.0 Tokens)
    - BEA Aqualogic 2.5 (on SUSE Linux)
    - Oracle ESB Suite 10.1.3.1(on Windows 2003)
  - Prototyped multi-vendor ESB interaction and cross-platform operations
  - Enabled legacy C/C++ code with gSOAP 2.7
  - .NET and J2EE Interoperability
  - Performance with large messages
  - Security (Multi-Domain)

Installed, configured, and benchmarked ESB products from major SOA vendors

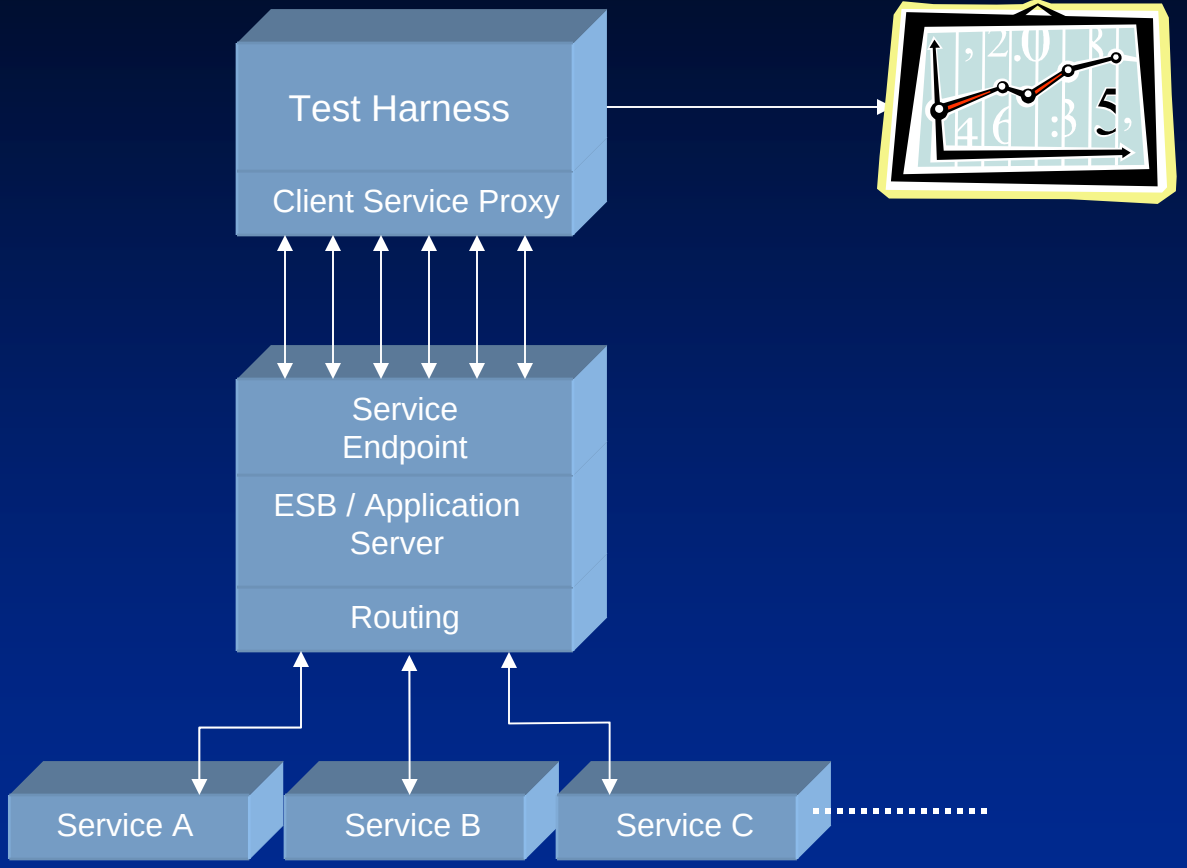
Prototype SOA Infrastructure

Integrated Systems & Solutions



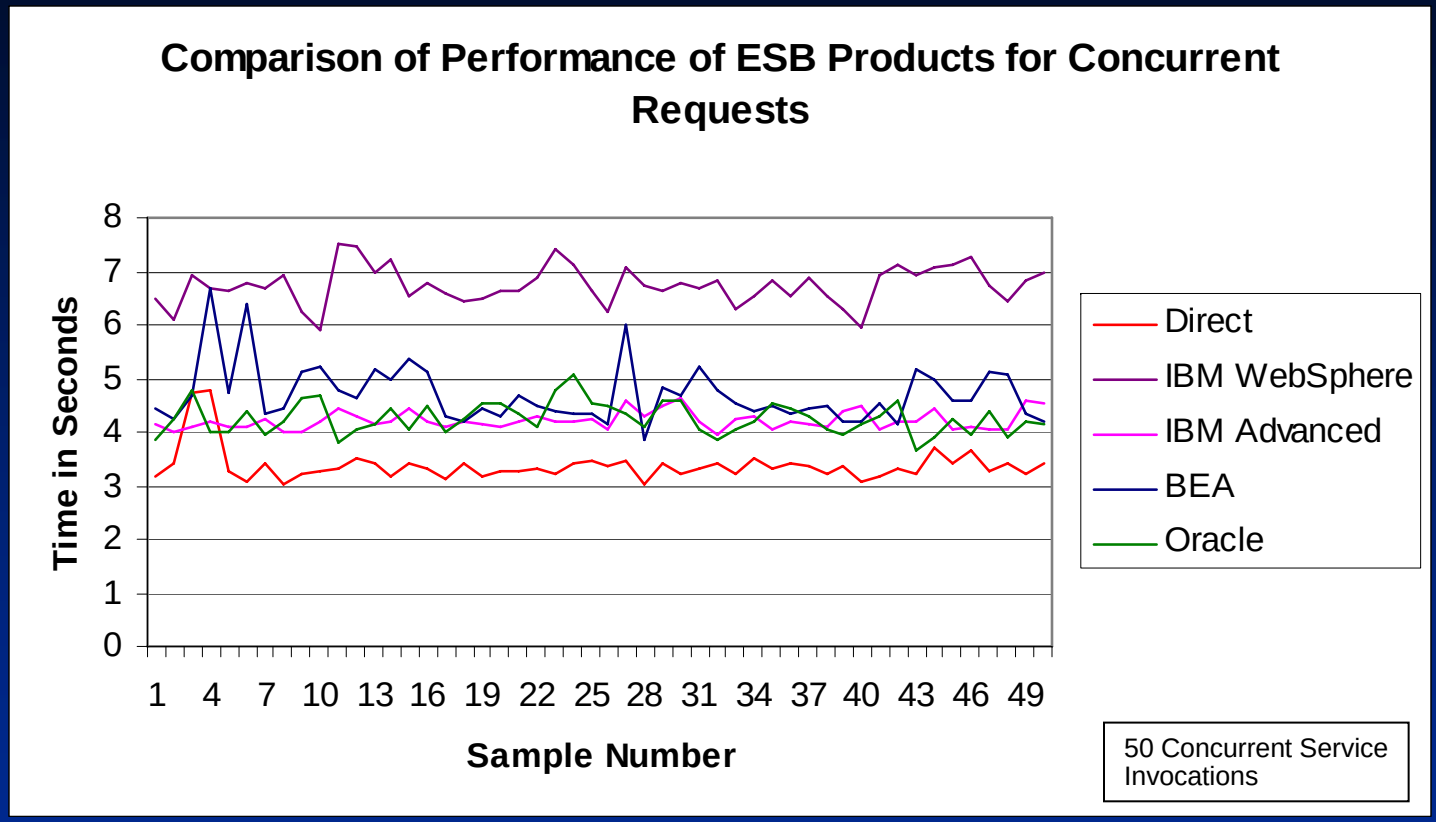
Prototype SOA Infrastructure

Constructed web services
"Test Harness" to generate both sequential and concurrent service invocations



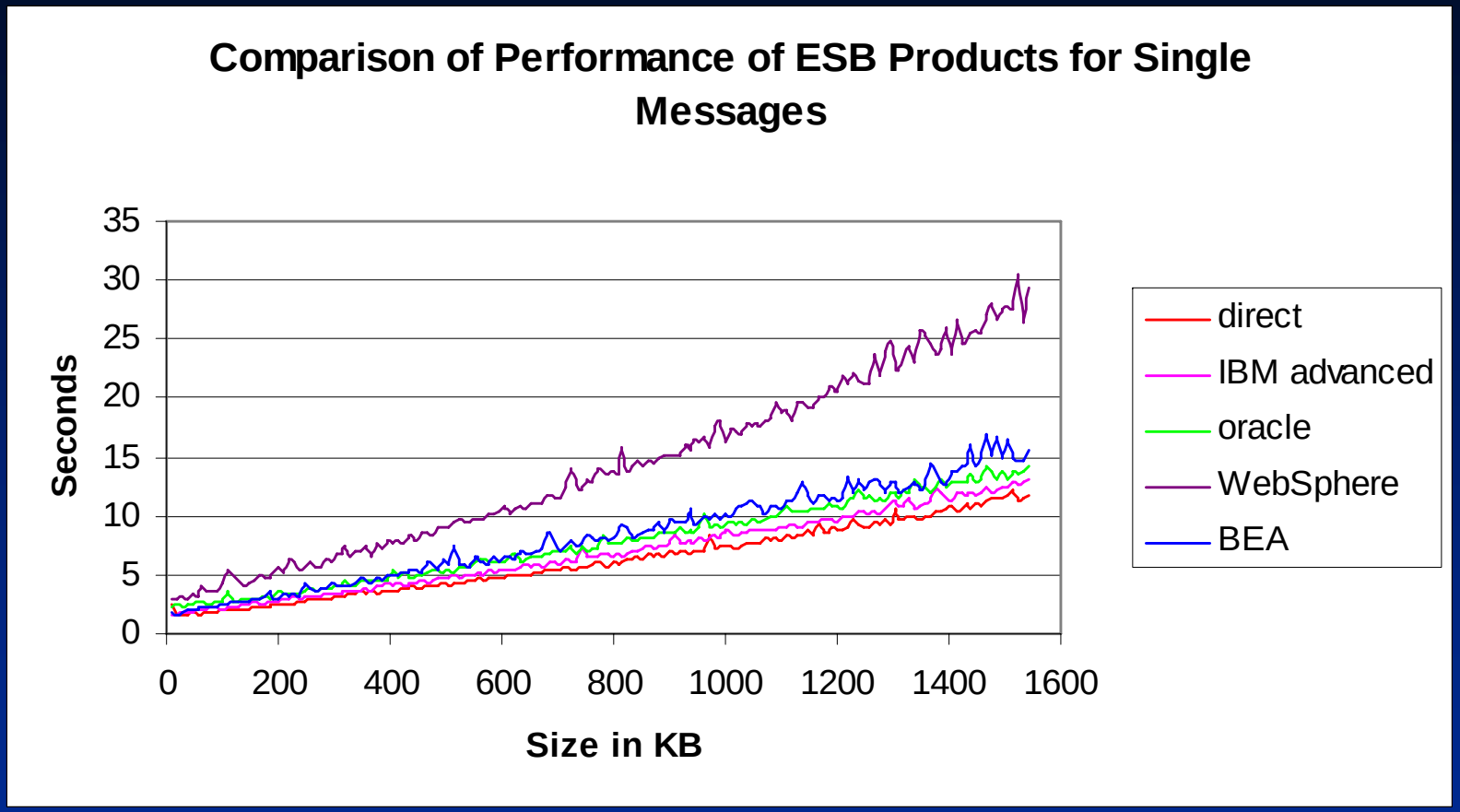
What additional overhead does an ESB carry compared to direct service invocation?
What are the performance limiting factors in using web services?

Implementing a Ground SOA



All major ESB products handle concurrent requests relatively well – some variability due to dynamic "garbage collection"

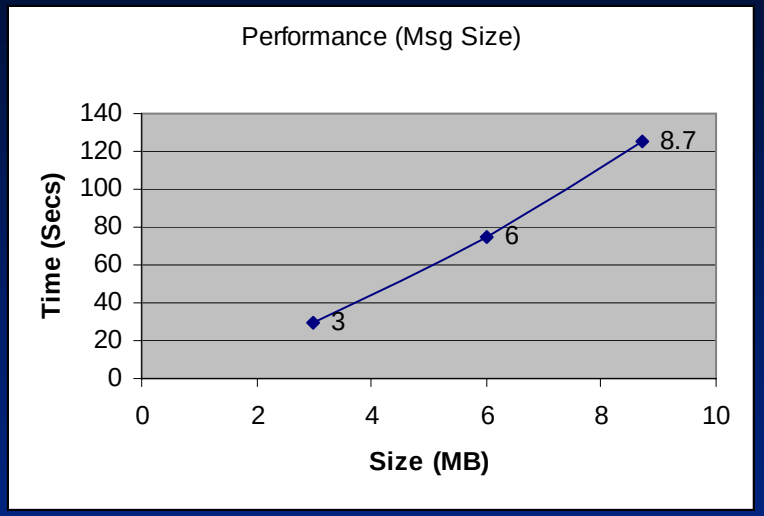
Implementing a Ground SOA



ESB overhead shows small but increasing overhead as message size increases – compared to direct service invocation

Web Services Performance and Large Messages

- IBM Advanced ESB and DataPower tested with large messages
  - 3MB (335 hrs Eph Data) response:
    - 30 seconds
  - 6MB (675 hrs Eph Data) response:
    - 75 seconds
  - 8.7MB (1080 hrs Eph Data) response:
    - 125 seconds



- Real Performance Bottleneck is in SOAP Processing for large messages
  - XML Serialization in client + XML De-serialization in server
    - CPU time and memory intensive
  - Message Size/Complexity dependent
- Additional ESB overhead can handle these size messages…
  - But heap size and timeout must be increased

Addressing Large Messages in Web Services

- Message Transfer Options
  - SOAP
    - Ordinary XML encoded payload document in SOAP envelope
  - SOAP with Attachments (SwA)
    - Compound Document Structure
    - MIME Encoding (De-facto usage standard) of attachment information
    - DIME Encoding (Direct Internet Message Encapsulation) largely obsolete
  - MTOM (Message Transmission Optimization Mechanism, W3C), XOP (XML-binary Optimized Packaging, W3C)
    - Relatively new standards
  - Out-of-Band Transfer
    - E.g., pass URI and use other transfer mechanism (FTP)
    - Places burden of decoding message payload back to the application

PREVIOUS DATA RESULTS

Alternatives exist to mitigate performance bottlenecks of XML Serialization/De-serialization with large messages

Web Services Performance Findings

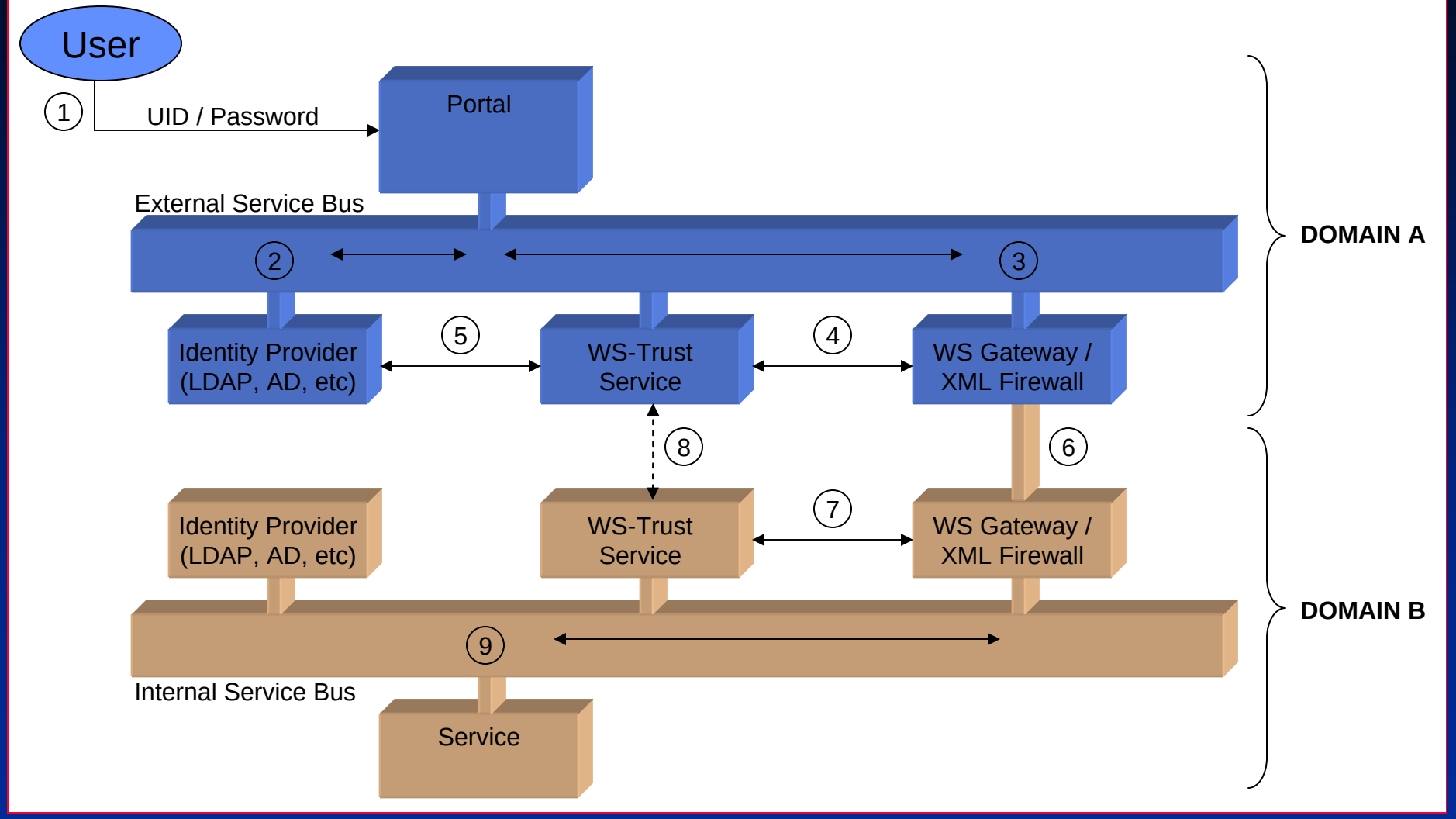
- Additional Performance Considerations
  - SOAP Encoding Styles
    - RPC/Encoded (Worst Performance)
      - *Deprecated and not WS-I compliant*
    - RPC/Literal (Middle)
    - Document/Literal Wrapped (Best Performance)
      - *Greater user control of parsing*
      - *Namespace element tagging allows complex datatype validation*
- Performance Conclusions
  - True performance bottlenecks due to XML Serialization and De-serialization of large messages—mitigate via:
    - Alternative transfer mechanisms (e.g. SwA) for XML payload messages > ~ 10 MB
    - SOAP encoding style (Document/Literal)
    - H/W appliance XML Accelerators
  - ESB products add modest overhead which increases as message sizes grows

Web Services and SOA Security

Multi-Domain SOA Security

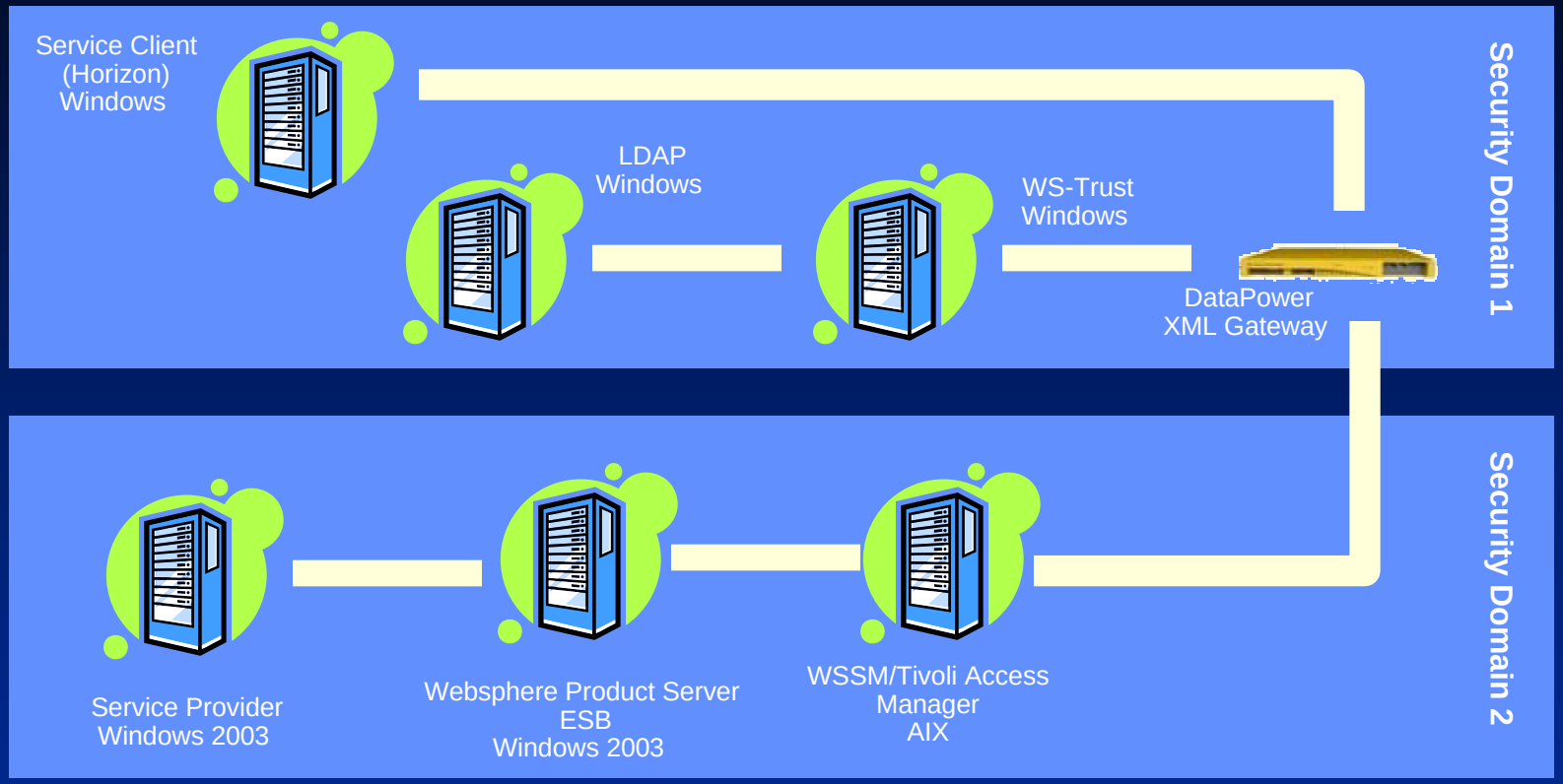
- Do not confuse Multi-Domain Security with Multi-Level Security (MLS)
- Multi-Level Security implies a single domain with electronic access by one or more individuals not briefed at all security levels (or compartments) for data within the system
  - Typically requires DCID 6/3 PL-4 protection
- Multi-Domain Security implies there are multiple infrastructure domains, managed by different organizations, that may have different requirements and standards
  - Data in each domain may in fact have same set of Classification Level(s), Compartments
  - Key issue is trust: you must trust that your partner's security implementation is reliable

Web Services and SOA Security: Logical Architecture



Web Services and SOA Security: Configuration 1

Integrated Systems & Solutions

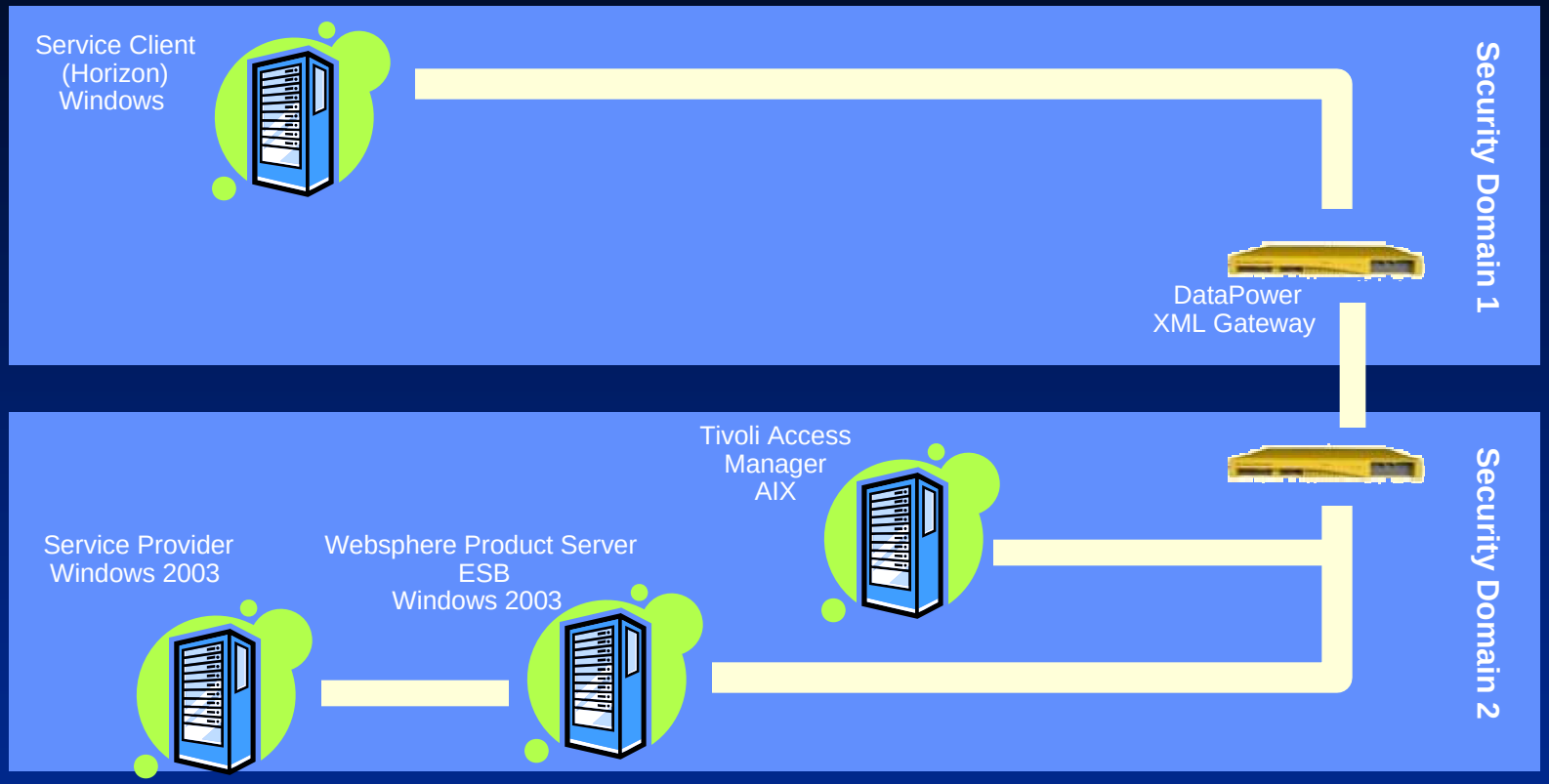


- Complex and fragile architecture but acceptable performance
- Componentized architecture permits flexibility
- TFIM implementation of WS-Trust and WSSM is still maturing
- Enforcement via WS ESB is proprietary; no security on response

SAML Token Exchange Across security domains

Web Services and SOA Security: Configuration 2

Integrated Systems & Solutions

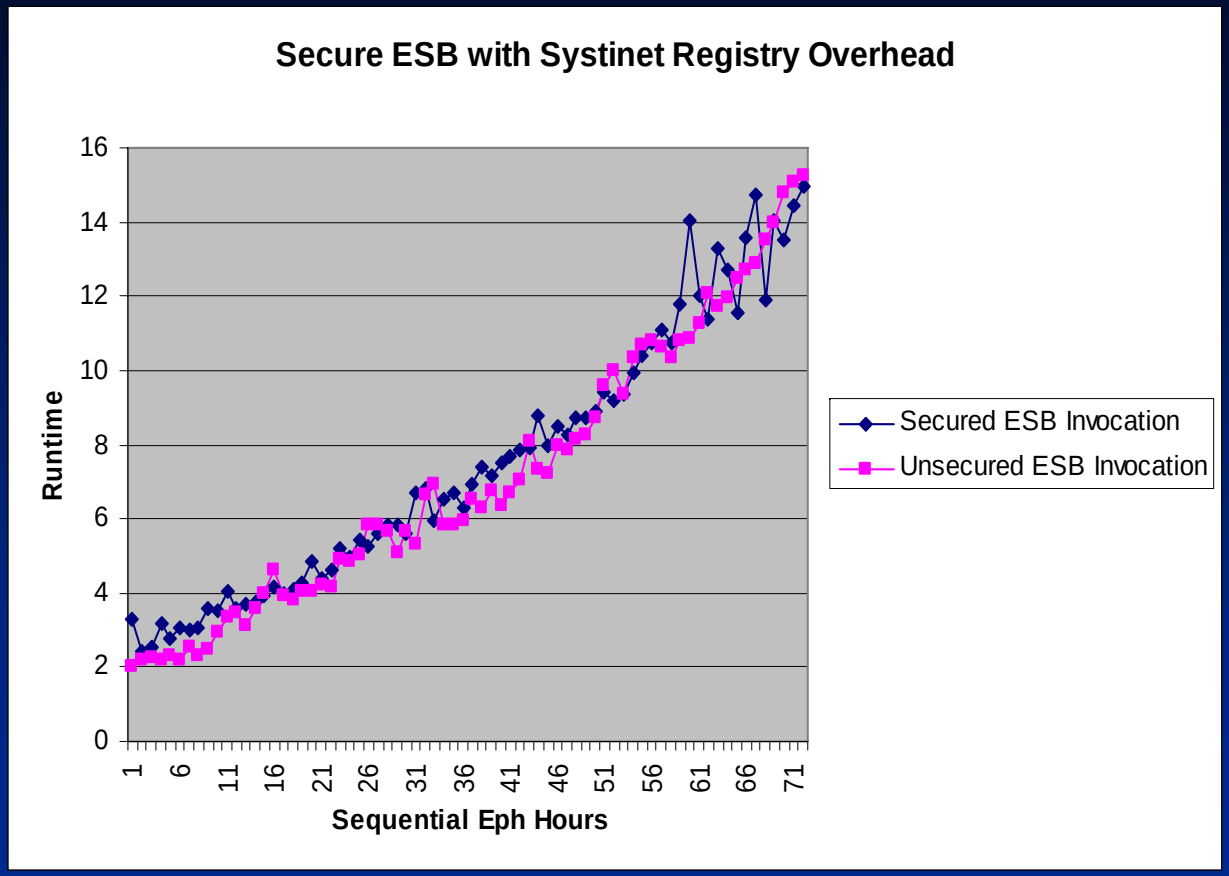


- Simplified and easy to configure; very fast
- Can transform and route messages based on content & policy
- Can sign and encrypt responses
- XML gateway product is proprietary
- Transformations can only be written in XSLT (no custom adaptors)

SAML Token Exchange Across security domains

Performance Overhead – Dynamic Routing & Security

- The two tests are identical (1-72 hours of Ephemeris Generation/Retrieval)
- Adding security (DataPower appliance, SAML 2.0 token generation, trust chain, etc.) and dynamic routing did not significantly degrade performance



Adding security and dynamic routing to service invocation did not dramatically alter performance

Summary and Conclusions

- SOAP/XML based web service performance is largely a factor of serialization and de-serialization of XML messages at mediation
  - Size of response is the critical factor in performance analysis: large size (MB range) results in rapid performance degradation
  - Alternative approaches for transferring large messages/files via web services required and available
    - Impacts how you should structure your services
- Additional ESB overhead is small compared to message size effect
  - ESB products handle consistent, moderate loads dramatically better than sudden, heavy loads
- SOAP encoding style has an impact: prefer document/literal wrapped
- Practical message size is not changed with the addition of cross-domain security
  - Additional network hops, but small data size exchanges in each

"A man has got to know his (COTS) limitations."