Zach Arlen
Application Developer

Back to the Future:

A Young Developer Explores the History of Application Integration

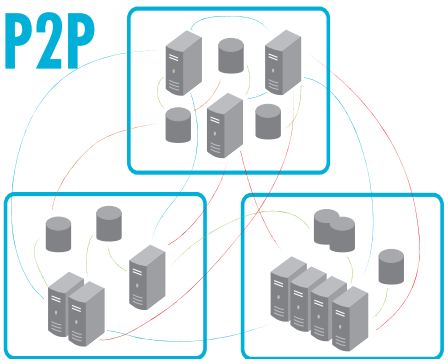
I've been out of college a few years now, and my very wise mentors often remind me that I have it pretty easy compared to when they were up-and-coming developers back in the day. They ramble on about punch cards, hanging chad and how much better the music was in the 80's. Since applications don't code themselves and systems don't naturally integrate with one another, I sat down with one of my, uh, more "experienced" mentors to find out why it is that things are so much "easier" these days.

I got a bit of a history lesson on application integration, and after hearing about the point to point, broker technologies, and web services/SOA approaches to integration, I've come to realize that things might actually be a little easier these days.

Back in the day…

Point to Point

The earliest of integration approaches, I'm told, began in the early days of application development. Back then, apps were monolithic and built with procedural-based languages, which inhibited the development of one-to-one relationships between applications and business processes. To bridge that gap, the Point-To-Point (P2P) method of systems integration was developed.
At first, P2P integration appeared to be an optimal solution. But, P2P integration had a limited ability to adapt to enterprise system growth, because it couldn't handle the high volume of custom interfaces. This led to a poorly maintainable and highly complex enterprise system, ultimately resulting in an integration approach shift.



How things were back in the day…

Broker Technologies

The second integration approach we discussed was broker technologies. Arising out of a new paradigm, "System of Systems," one of the first broker technologies widely implemented was Message Oriented Middleware (MOM). MOM promoted inter-system communication by placing message broker software between two disparate systems.

Web Services/SOA

The necessity for a global messaging specification across systems promoted a shift to web services (WS). WS allows for the integration between systems using previous system design patterns. But, unlike previous brokers, it shored up loose ends in standardization by implementing a global messaging specification, known as SOAP. WS intrinsically abstracts systems into unassociated units of functionality, leading to the decoupling of systems and simplification of the enterprise system structure. Simply put, the aptly named Service-Oriented Architecture (SOA) is an orchestration of defined WS into new business processes.

…Back to the Future - Integration the Easy Way

Web services and SOA bring us to the current historical moment of systems integration. With the introduction of the SOAP standard, WS really reduced integration headaches. But, since I work for an Oracle Partner, the story of why it is that I have it easier as a developer has only been half told. A major, scratch that, THE reason I have it so much easier today is because I get to implement Oracle's industry-leading technologies. Oracle BPEL Process Manager, Oracle Service Bus, Oracle Business Process Management, Oracle Data Integration and Oracle Virtual Directory take the pain out of integration.

BPEL Process Manager provides a robust set of tools for service integration. Today, BPEL is emerging as the industry standard for orchestrating services into an end-to-end process flow, reducing the cost and complexity of process integration. Oracle BPEL Process Manager enables companies to assemble processes in a standards-based manner and delivers crucial functionality to develop towards a Service-Oriented Architecture. As well, BPEL Process Manger is "hot-pluggable", allowing business to deploy Oracle AIA products into existing environments.

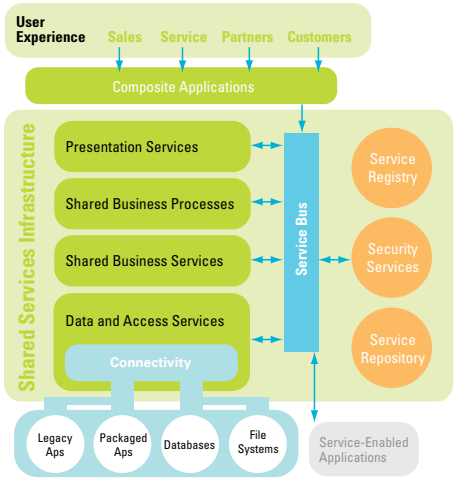
Oracle Service Bus is a scalable application integration platform that delivers high volume performance. This technology is designed to link, mediate and direct interactions between multiple heterogeneous services, legacy systems, packaged applications and enterprise service bus instances across an enterprise-wide service network. Oracle Service Bus can be considered the back-bone for SOA messaging and offers stellar performance, scalability, reliability, manageability and interoperability with IBM, Tibco and Oracle messaging technologies.

Oracle Business Process Management (BPM) offers another platform for integration. For more details about Oracle BPM, see Elias Karthan's article, Oracle Business Process Management: Keeping You Ahead of the Curve, also in this issue.

Oracle Data Integration (ODI) provides users with a unified view of an aggregated set of data across several sources. ODI makes this process more manageable by providing a fully unified solution for building, deploying and managing complex data. Furthermore, ODI is a highly portable Java-based, SOA native platform with open metadata.

Oracle Application Integration Architecture (AIA) provides businesses greater agility using the applications already in use. In a competitive corporate environment, today's companies must adapt and evolve to ever changing business requirements. Oracle's AIA is an application independent framework that enables you to implement cross-application business processes on a SOA environment.

SOA using ESB



While applications still don't code or integrate themselves, I get it that developers have an easier time with integration today. All thanks to inventive developers, technological advancements and Oracle technology. Although I won't concede that today's IT professional has it easy, I'm definitely glad that we don't have to worry (as much) about integration.