Sign in

Enter keywords

ARCHITECTURE
&GOVERNANCE magazine

Home Sign Up Articles Current Issue Events Blog Resources Staff Contact Us

You are here: Home / The Hub and the Edge: Balancing the Responsibilities

The Hub and the Edge: Balancing the Responsibilities

Submitted by A-G Magazine on Sat, 2008-09-13 09:05

SHARE

Between the Edge and the Hub

Ever since two applications were first plugged together, architects have debated where integration logic belongs: The Hub or the Edge.

Hub supporters believe that central integration systems are the place to run integration logic. In the IT organization, Hub advocates are usually people with central coordinating roles, such as managers or integration architects–just the people who happen to control the Hub systems! And vendors of Enterprise Application Integration (EAI) and Enterprise Service Bus (ESB) products naturally advocate for the Hub.

Edge advocates, on the other hand, set the responsibility on the applications which are wired together, asking endpoint application teams to execute the integration, without specifying central systems. In the IT organization, managers of individual applications and of tactical integrations support this approach, as do vendors of Service-Oriented Architecture (SOA)support applications.

This article will offer a compromise based on the assignment of responsibility: Integration experts should focus on integration-specific functionality, and endpoint specialists should do what they do best, exposing the logic of their own application.

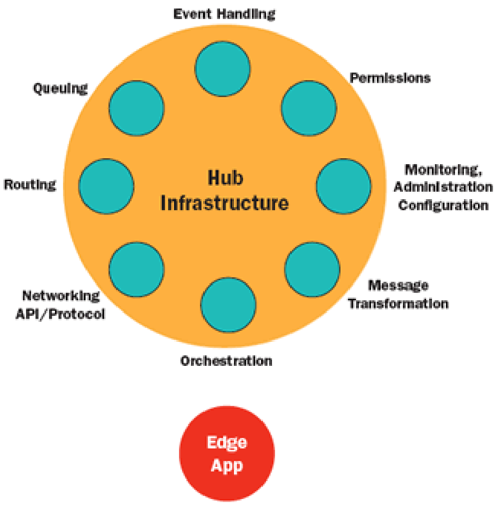
I'll illustrate the ideas with a simple example taken from the world of insurance, where the heterogeneity of legacy systems and the increase in B2B integration are rapidly driving the adoption of standard integration architectures.

Unbalanced Architectures

Hub-Only: All the Responsibility in the Center

When powerful Hub software does everything, you don't need to touch the end applications–everything is under the control of the Hub (figure 1).

Figure 1: All the Responsibility in the Hub



This approach eliminates code duplication across the endpoints, and makes it easier to interface with legacy or third-party systems. It empowers system managers, who can control and monitor the system at a central point, and reduces complexity, since each endpoint is spared the need to connect to multiple other endpoints (figure 2).

Yet managers of IT departments don't like dependency on a common Hub technology, while Hubfocused integrators find themselves chasing after changing application interfaces.

A Fortune 500 insurance company had over 150 applications participating in integrations. Although the chief architect attempted to introduce a commercial EAI hub in the mid-90's, he encountered significant pushback from the departmental IT managers in charge of the applications, who did not want to cede technical control to the proprietary system. Thus, integrators had to write adaptation code for each application by themselves, and when this proved too difficult, they

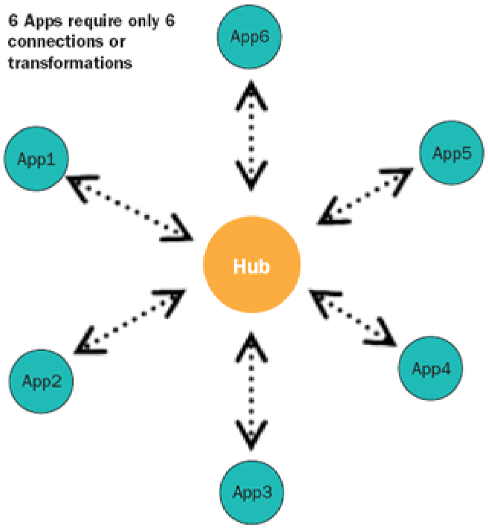
Popular Topics

Accenture alignment budgeting business alignment business architecture business capability model business intelligence business transformation CIO EA enterprise architecture executive leadership FEA Forrester futurism Gartner George Paras governance Infosys IT Governance IT Management IT planning IT spending it strategy Jonas Lamis leadership metrics opening thoughts Podcast Project management SOA standards strategic planning strategy survey visualize

more tags

would often revert to point-to-point implementations.

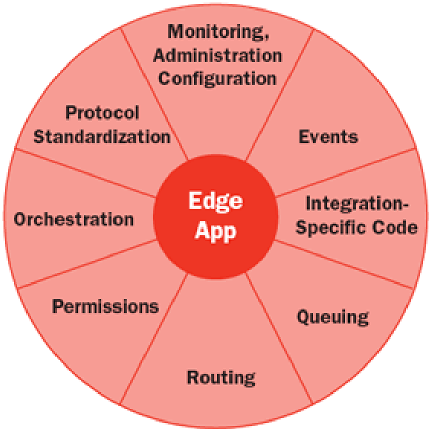
Figure 2: Deployment Simplicity Granted by a Hub



Edge-Only: All the Responsibility in the Endpoint Applications

In an Edge-based approach, the applications hold all integration functionality (figure 3). This allows independence from specific integration Hub, and avoids holy wars over proprietary Hub-centric protocols. SOA, with its emphasis on services exposed by endpoints, has popularized such architectures.

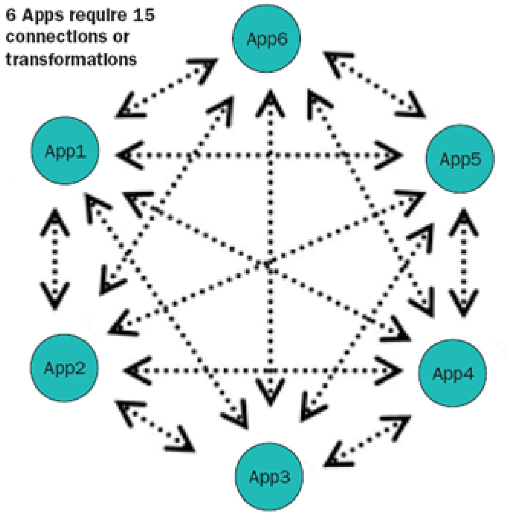
Figure 3: All the Responsibility in the Edges



In the Edge-based approach, teams responsible for applications or specific integrations are freed from dependence on central architects and managers. Application teams get the responsibility for exposing the functionality of their own products, and central architects need not be concerned with endpoint implementation internals.

Unfortunately, endpoint applications are not the best place for true integration logic. Complexity explodes when each endpoint must handle every type of interaction with every other endpoint (figure 4).

Figure 4: Deployment Complexity of Edge-Only Architectures



In the insurance organization, as difficulties mounted with the EAI system, integrations came to be developed point-to-point. As each application had to connect to other applications and to orchestrate sequences of service calls, integrations became impossible to manage and maintain.

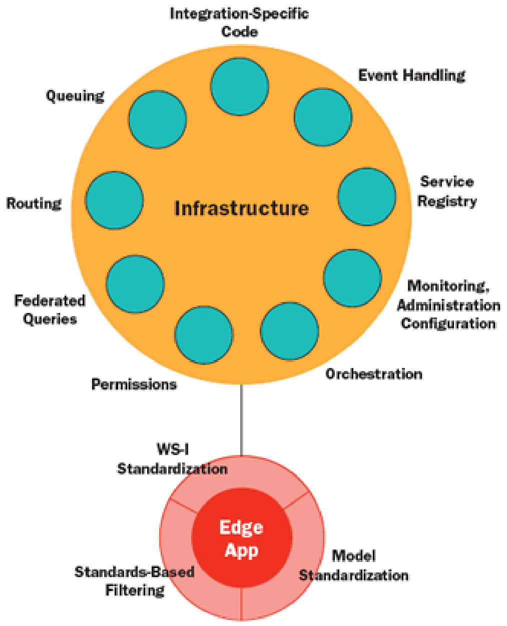
A Balanced Architecture

The key to good architecture is distinguishing what is likely to change from what is likely to stay the same, and distributing responsibility accordingly. Endpoint experts should implement features which depend on the application, yet stay the same in each integration scenario. On the other hand, integration experts should take care of functionality which varies per integration scenario, but does not depend on the applications' internals.

This balance can take on different forms in different types of IT organizations. Organizations with flexible in-house software, or independent software vendors (ISVs) sending integrated suites, have the luxury of inserting all needed functionality into their own product, and so tend more to the Edge. On the other hand, enterprises with a heavier weighting of commercial off-the-shelf software (COTS) must work with what they are given, and must connect to the application from the outside; these put more weight on the Hub But the basic principles remain the same: The Edges must balance responsibility with the Hub (figure 5).

In the insurance organization described, the emerging consensus for Web Services standards produced management support for the standardization of a service layer over legacy interfaces, reducing the overhead for protocol conversions. A balance between the Hub and the Edge evolved organically–the integration teams adopted an ESB to make their job more efficient, while endpoint teams continued refining their interfaces, which relieved them from repeated integration burdens. A balanced assignment of responsibilities emerged as the one way to do reusable, effective integration.

Figure 5: Responsibility Balanced between the Applications and the Infrastructure



The Applications

In the balanced approach, Edge teams have the responsibility to expose internal systems using shared standards. The adapters are closely connected to the application, giving them access to internal APIs where available.

All services must conform to the basic Web Services Interoperability standard, which has wide consensus and is easily implementable in multiple software platforms. They must structure their messages in a standard interchange format and register their interfaces into a service registry.

The Edge teams have total control over the implementation of the engines, which expose these standard interfaces. Managers are far easier to get on board when they maintain their autonomy. The coordinating architect provides design guidelines and reference implementation based on their experience, without constraining the endpoints' implementation design.

The Infrastructure

The integration team has the responsibility for integration-support functionality. Those aspects of integration that vary per scenario and do not depend on the innards of the endpoint applications, such as routing, orchestration, message security, and scalability as well as the design, development, management and monitoring of integration flows.

When the infrastructure does the heavy lifting, application teams no longer need to be experts in these highly technical areas.

The integration infrastructure is not necessarily a monolithic EAI server. Today's ESBs offer distributed containers that can spread integration functionality across multiple nodes as necessary for scalability, security, or distribution of responsibility. Moreover, best-of-breed systems can offer each of the specialized integration functionalities: an orchestration, an identity server, a message queue, and so on.

Common Interchange Formats

When enterprise software systems expose interfaces in their own proprietary schemas, every new integration project requires new business analysis and coding of transformations.

The solution for this chaos is well known. Industry consortia, such as FIX in equities, RosettaNet in electronics, and ACORD in insurance, determine standard messages to be passed between participants in integration flows, hiding implementations. These industry-wide standards are successfully connecting disparate organizations today, in large-scale internal and B2B integrations.

Within each enterprise, central architects define schemas that support their organization's needs, starting with relevant industry standards and including input from the various stakeholders. They then steward their ongoing development. They must orient the schemas to the concepts of the business, rather than the technical specifications of endpoints. To keep all stakeholders aligned, they publish the schemas to a service registry.

The endpoints expose interfaces in this common format, transforming to and from their application specific formats.

With the registry, the central architects then track conformance to the common format. To achieve this, they must get executive sponsorship of a governance process which brings together managers and architects from the central team, from integration teams, and from the endpoint applications.

The insurance company had the benefit of XSD Schemas from the ACORD standard. These served for B2B integrations. For internal projects, the central architecture team defined extensions. In this way, well-understood interfaces based on insurance industry concepts replaced proprietary data structures, which had incomprehensible and poorly defined fields. Each application team had to implement the translation to and from the standard only once.

Making It Happen

The architects and managers of the central team are in charge of creating and maintaining this balance. Their role goes beyond technical development and support. They also provide vision and executive direction for the organization's integration architecture, drilling down to clear and consistent decisions on infrastructure, such as ESBs and standard protocols. They then set policies and standards, guiding the organization in adopting these decisions.

This can only happen when central architects explain the value of their architecture vision to the endpoint and integration teams, and central managers track and verify conformance. By building and leading governance processes and committees, and with the aid of service registries and other tools, architects and management assure that the endpoint teams expose interfaces according to the standards, that the integration teams provide the integration functionality, and more generally, that each role in the system is being fulfilled.

It is impossible to adopt this architecture in a single step. It must be adopted gradually, proving value and feasibility on each step, starting with a single integration project. The best integrations to start with are event-based, workflow-oriented integrations, as opposed to high-throughput or data synchronization tasks. It's also easier to get green-field projects to comply than ongoing development of existing integrations. The integration should include three or more participating applications in order to prove the generality of the approach.

At this stage, the endpoint teams create a small set of standards-conformant Web services, targeted directly at the project's needs, layering them over existing APIs. The integrators use these APIs and their integration-infrastructure tools to wire the applications together. Not all the infrastructure comes into play at this early phase, only enough to meet the needs of the project and prove the architecture's benefits. Throughout the process, the central architect and the infrastructure-systems team contributes directly to the project to speed its progress and to closely support development. Indeed, at this point they still have the flexibility to iteratively change the infrastructure and model as necessary.

The endpoint teams will soon see the benefits in reduced support costs, and the integrators will see that they can efficiently create reusable integrations without getting into application internals. The executive leadership, which has a higher-level view of the process, will see the greatest benefit, as previously uncoordinated teams align on a common vision of balanced responsibilities.

Conclusion: Finding your Balance

Integration architects often get mired in political debates: Who has control, the central integration team or the managers of endpoint applications? The correct compromise applies criteria of expertise and responsibility, balancing the roles of the Edge applications with that of Hub systems. When the central architects and managers define the technical architecture and roles, then govern the ongoing process, IT organizations can create a well-defined and flexible architecture, align their organization, and achieve quality, speed, and maintainability in integration development.

Bibliography

Enterprise Service Bus: Theory in Practice, David Chappell, 2004.

"Central Information Models for Data Transformation," Joshua Fox, *eAI Journal/Business Integration Journal*, May 2003.

"Enterprise Semantics: Aligning Service Oriented Architecture with the Business," Joshua Fox and Joram Borenstein, *Web Services Journal*, May 2005.

***Joshu Fox** (www.joshuafox.com) is an Israel-based software architect and a frequent writer and speaker in his field. Now with Hewlett Packard, his experience includes industrial Semantic Web systems, as well as high-performance Internet collaboration servers. Fox's educational background includes a PhD from Harvard and a BA summa cum laude from Brandeis University.*

Tags: Articles, EAI, ESB, Issue 3-2, SOA