Hype Cycle for Open-Source Software, 2009

Mark Driver, Massimo Pezzini, Nikos Drakos, Andreas Bitterer, Ted Friedman, Andrea Di Maio, David Gootzit, Yefim V. Natis, Bob Igou, Thomas E. Murphy, Eric Knipp, Michael A. Silver, Annette Jump, Donald Feinberg, Milind Govekar, Cameron Haight, Mick MacComascaigh, Mark R. Gilbert, W Schulte, John R. Phelps, Mike Chuba, Michael Maoz, Jim Duggan, George J. Weiss, Donna Scott, David Norton, Brian Gamage, Monica Basso, Daniel Sholler, Jess Thompson, Andrew Butler

Use the open-source software Hype Cycle to measure the maturity and market saturation of open-source solutions across the IT industry.
TABLE OF CONTENTS

Analysis ................................................................................................................................. 4
  What You Need to Know .................................................................................................... 4
  The Hype Cycle .................................................................................................................. 4
  The Priority Matrix ......................................................................................................... 6
On the Rise .......................................................................................................................... 7
  Open-Source SaaS/Cloud-Enabled Application Platforms .................................................. 7
  Open-Source AD OOA&D Tools ....................................................................................... 9
  Open-Source Public-Sector Vertical Applications ........................................................... 10
  Open-Source Technology for Extreme Transaction Processing ........................................ 11
  OSGi for Application Infrastructure Middleware .......................................................... 13
At the Peak .......................................................................................................................... 16
  Open-Source Data Integration Tools ................................................................................. 16
  Open-Source Service Management Tools ....................................................................... 17
  Open-Source Virtualization Platforms ............................................................................ 18
  PC Linux for Consumers (Mature Markets) ....................................................................... 19
  Open-Source Content Management ................................................................................ 19
  Open-Source Horizontal Portals ..................................................................................... 21
  Open-Source AD Testing Tools ........................................................................................ 22
  Open-Source Application Life Cycle Management Tools ................................................ 23
  PC Linux for Consumers (Emerging Markets) ................................................................. 24
  Open-Source Business Intelligence Tools ....................................................................... 25
  Open Source for CRM ...................................................................................................... 26
Sliding Into the Trough ...................................................................................................... 27
  Open-Source Message-Oriented Middleware .................................................................. 27
  Open-Source Enterprise Service Bus ............................................................................... 28
  Linux on Desktop for Mainstream Business Users ........................................................... 30
  Open-Source Social Software .......................................................................................... 31
  Open-Source DBMS ......................................................................................................... 32
  Open-Source Office Products ......................................................................................... 34
  Linux on 16 to 64 Processors .......................................................................................... 35
Climbing the Slope ............................................................................................................. 36
  Spring ............................................................................................................................... 36
  IT Services for Open-Source Software ............................................................................ 38
  Mission-Critical Workloads on Linux ............................................................................. 40
  Linux on System z ............................................................................................................ 41
  Open-Source Java EE Application Servers ...................................................................... 43
  PC Linux for Data Entry Workers ................................................................................... 44
Appendixes .......................................................................................................................... 46
  Hype Cycle Phases, Benefit Ratings and Maturity Levels ................................................ 48
Recommended Reading ...................................................................................................... 49

LIST OF TABLES

Table 1. Hype Cycle Phases ................................................................................................. 48
Table 2. Benefit Ratings ..................................................................................................... 48
Table 3. Maturity Levels ...................................................................................................... 49
LIST OF FIGURES

Figure 1. Hype Cycle for Open-Source Software, 2009................................................................. 5
Figure 2. Priority Matrix for Open-Source Software, 2009 ............................................................ 7
Figure 3. Hype Cycle for Open-Source Software, 2008................................................................. 46
ANALYSIS

What You Need to Know

In 2009, open-source software (OSS) is an integral element of many IT markets; and in others, it is just emerging as a viable alternative to traditional closed-source solutions. As we've seen in years past, the rate of maturity and level of saturation varies from market to market. In several segments, open-source solutions rank among the industry-leading products; but, in other market segments, open-source solutions remain outside the scope of all but the most aggressive technology adopters. Most mainstream IT organizations find it impractical to avoid OSS and, instead, are focusing on managed strategies that blend the best of open-source and closed-source solutions. All IT organizations must create standards, practices and management policies that take into account the ubiquity of OSS. Toward this end, IT maturity must learn to closely assess the maturity and business value of open-source projects on a case-by-case basis. Gartner's Hype Cycle model is one tool for enterprises to use to streamline and simplify this effort.

The Hype Cycle

As we saw in 2007 and 2008, mainstream OSS adoption patterns continue to shift in two ways:

- Open-source solutions are being deployed in increasingly mission-critical scenarios, where the service level must be equal to or better than closed-source alternatives.
- Open-source solutions are being adopted by increasingly conservative IT organizations that regard cost and risk mitigation as their primary concerns.

The vendor positions in this Hype Cycle reflect these patterns, because technologies emerging from the Trough of Disillusionment are best-suited to both challenges.

In 2009, we see open source continuing to play a crucial role in new IT initiatives. For example, open source is already a cornerstone of emerging efforts around social computing (see "Open Source in Social Computing, 2008") and advanced transactional systems (see "Open Source in Extreme Transaction Processing, 2008"). Open source will play a crucial role in cloud computing as well, as it becomes the dominant infrastructure for IT cloud infrastructures during the next several years.

This year, a number of profiles have made large jumps forward (e.g., application development testing tools, enterprise service bus, Linux on 16 to 64 processors and DBMS). In each of these cases, we've seen significant acceleration in interest and adoption during the past 12 months (see Figure 1).
Figure 1. Hype Cycle for Open-Source Software, 2009

expectations

PC Linux for Consumers (Mature Markets)
Open-Source Virtualization Platforms
Open-Source Service Management Tools
Open-Source Data Integration Tools
OSGi for Application Infrastructure Middleware
Open-Source Technology for Extreme Transaction Processing
Open-Source Public-Sector Vertical Applications
Open-Source AD OOA&D Tools
Open-Source SaaS/Cloud-Enabled Application Platforms

Open-Source Content Management
Open-Source Horizontal Portals
Open-Source AD Testing Tools
Open-Source Application Life Cycle Management Tools
PC Linux for Consumers (Emerging Markets)
Open-Source Business Intelligence Tools
Open Source for CRM
Open-Source Message-Oriented Middleware

Open-Source Enterprise Service Bus

Linux on Desktop for Mainstream Business Users
Open-Source Social Software

Open-Source Enterprise Service Bus

As of July 2009

Technology Trigger
Peak of Inflated Expectations
Trough of Disillusionment
Slope of Enlightenment
Plateau of Productivity

Years to mainstream adoption:
● less than 2 years
○ 2 to 5 years
● 5 to 10 years
△ more than 10 years
★ obsolete before plateau

Source: Gartner (July 2009)
The Priority Matrix

Continuing a trend from 2008, the 2009 OSS Hype Cycle represents a relatively even mix of high, moderate and low priority technologies: This is evidence of open source’s impact on software markets as an evolution, rather than a revolution. The vast majority of mainstream IT organizations are adopting open source in a blended strategy, mixing open-source and closed-source solutions within the larger enterprise software portfolio. OSS also plays a transformational role in innovative IT initiatives, such as extreme transaction processing, establishing itself as a cornerstone in the evolution of distributed computing.
### Figure 2. Priority Matrix for Open-Source Software, 2009

<table>
<thead>
<tr>
<th>Benefit</th>
<th>Years to Mainstream Adoption</th>
<th>Low</th>
<th>Moderate</th>
<th>High</th>
<th>Transformational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Benefit</td>
<td>less than 2 years</td>
<td>Open-Source AD OOA&amp;D Tools</td>
<td>Open-Source Testing Tools</td>
<td>Open-Source AD Testing Tools</td>
<td>Open-Source Public-Sector Vertical Applications</td>
</tr>
<tr>
<td>Benefit</td>
<td>2 to 5 years</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td>Open-Source Horizontal Portals</td>
<td>Open-Source Service Management Tools</td>
<td>Open-Source Technology for Extreme Transaction Processing</td>
</tr>
<tr>
<td>Benefit</td>
<td>5 to 10 years</td>
<td>Open-Source Service Management Tools</td>
<td>Open-Source Virtualization Platforms</td>
<td>Open-Source Application Life Cycle Management Tools</td>
<td></td>
</tr>
<tr>
<td>Benefit</td>
<td>more than 10 years</td>
<td>Linux on System z Mission-Critical Workloads On Linux</td>
<td>Open-Source DBMS</td>
<td>Open-Source Application Life Cycle Management Tools</td>
<td></td>
</tr>
<tr>
<td>Benefit</td>
<td></td>
<td></td>
<td>Open-Source Enterprise Service Bus</td>
<td>PC Linux for Consumers (Emerging Markets)</td>
<td></td>
</tr>
<tr>
<td>Benefit</td>
<td></td>
<td></td>
<td>Open-Source Message-Oriented Middleware</td>
<td>Open-Source Application Life Cycle Management Tools</td>
<td></td>
</tr>
<tr>
<td>Benefit</td>
<td></td>
<td></td>
<td>Open-Source Social Software</td>
<td>PC Linux for Consumers (Emerging Markets)</td>
<td></td>
</tr>
<tr>
<td>Benefit</td>
<td></td>
<td></td>
<td>OSGI for Application Infrastructure Middleware</td>
<td>Open-Source Application Life Cycle Management Tools</td>
<td></td>
</tr>
<tr>
<td>Benefit</td>
<td></td>
<td></td>
<td>Spring</td>
<td>Open-Source Application Life Cycle Management Tools</td>
<td></td>
</tr>
</tbody>
</table>

**As of July 2009**

**Source:** Gartner (July 2009)

**On the Rise**

**Open-Source SaaS/Cloud-Enabled Application Platforms**

**Analysis By:** Yefim Natis
**Definition:** A software-as-a-service (SaaS)-enabled application platform (SEAP) is a programmable platform for the development and deployment of custom applications that is specially designed to support SaaS-style applications. To natively support SaaS applications, a SEAP must enable multiple user organizations (tenants) to deploy customized application instances in one running instance of the platform with full security, privacy and performance isolation (see "Introducing SaaS-Enabled Application Platforms: Features, Roles and Futures"). Other requirements, which are common to all cloud software, include elastic scalability and fine-granularity usage tracking.

In principle, SEAPs are a variant of an application server; however, to meet the SaaS-enabling requirements, a separately standing and designed technology stack is typically required. Also, because of the application customization requirements and the lack of standardization, in most cases, SEAP includes dedicated development tooling. SEAP enables the advanced shared-everything model of cloud computing (see "Reference Architecture for Multitenancy: Enterprise Computing in the Cloud").

It is important to differentiate application platform as a service (APaaS), which is an off-premises service, from SEAP, which is an on-premises technology category. Simply speaking, SEAP is an application server with some special features, such as multitenancy, that make it suitable for SaaS-style applications. To use it, you need to buy it, just as you would any other application server. APaaS is a service, rather than a product. Someone else runs the software, you just pay for membership and get access to a running environment where you can develop and deploy applications. To offer APaaS, the provider must have developed or bought an SEAP.

Most SEAP users are independent software vendors (ISV) that build SaaS-style business applications and are looking for SaaS-enabling platform technology. The more technically advanced ISVs value the advantages of an open-source license for such technology. This enables them to make the acquired platform technology their own by seeing its internal code and having the ability to change it to meet their specific needs.

**Position and Adoption Speed Justification:** SaaS-style applications (such as CRM) are typically offered by vendors that have developed proprietary enabling-platform technology. Thus, the SEAPs under such applications as Oracle CRM (Siebel) or Workday are not visible as separate entities. However, as application vendors want to attract an ecosystem of ISV partners, they are beginning to open their SEAPs for third-party use in the form of a dedicated APaaS (NetSuite NS-BOS is an example). This trend recognizes the existence of the differentiated platform technology for SaaS-style applications (development and deployment). Not all vendors come to offering an SEAP from a SaaS application background. Some offer SEAP in the manner of an application platform with special characteristics without also offering their own live APaaS (Magic Software Enterprises, Cordys and Rollbase are example vendors). LongJump, Archer Technologies, Qrimp and Bungee Labs offer both SEAP and APaaS.

Although the number of offerings is increasing, the market recognition of APaaS is only emerging. However, as SaaS-style application development moves to center stage of business innovation, SEAP technology will become an essential enabling technology for SaaS ISVs, APaaS providers and some IT organizations as well. The open-source SEAP offerings lag behind the general SEAP offerings in adoption and availability. Few of the current SEAP vendors offer an open-source license for their technology, and those that do, in some cases, offer only a subset of the full platform as open source.

**User Advice:** Most early buyers of SEAPs are ISVs that plan to create their own SaaS application stacks and, perhaps, create their own partner ecosystems. Only the largest IT users would be interested in building their own internal private-cloud environments — thus, possibly requiring an internal SEAP. For these largest users, as well as the ISVs, the choice of an SEAP is difficult and carries some risks, including the uncertain viability of most providers and the threat of
locking the organization into a proprietary programming model of the chosen SEAP. Nonetheless, developing an SEAP in-house is likely to carry some of the same risks, and at a higher cost. An open-source SEAP provides some protection against these risks, as it reduces the vendor lock-in.

ISVs and IT organizations planning to build SaaS-style applications should evaluate the SEAP options before plunging into developing their own platform technologies. When selecting an SEAP, ensure access to the product's source code, in case of problems with the vendor. An open-source option might be the optimal offering for technically skilled ISVs. Organizations looking to use third-party SaaS-style business applications should inquire about the nature and capabilities of the underlying SEAP to determine the SaaS provider's ability to attract ISV partners, as well as the user's ability to customize and extend the SaaS application.

**Business Impact:** As mainstream organizations begin to trust and appreciate the benefits of cloud computing, the now-overriding reliance on application servers as the foundation for business applications will begin to shift to their new incarnation — the SEAP. SEAP is the enabling technology for APaaS and SaaS-style cloud applications. It is fundamental to the software industry transition from a business model based on selling software products to IT users to new business models based on selling software-based services to IT organizations. Vendors with the most-advanced SEAPs will achieve SaaS-style application services of the highest technical quality. Vendors that can establish a standard in the programming model for an SEAP (which is now lacking) will have a major advantage over their competitors. Vendors offering an open-source licensing option will attract more advanced ISV deployments, but may have greater challenges in generating long-term revenue.

**Benefit Rating:** Transformational

**Market Penetration:** Less than 1% of target audience

**Maturity:** Emerging

**Sample Vendors:** LongJump; Rollbase; WaveMaker Software

**Recommended Reading:**
- "Application Infrastructure for Cloud Computing: An Emerging Market"
- "Key Issues for Open-Source Software, 2009"
- "Reference Architecture for Multitenancy: Enterprise Computing 'in the Cloud'"
- "LongJump Reality Check: Product vs. Service in the Early Cloud Age"
- "Key Issues for Cloud-Enabled Application Infrastructure, 2009"
- "Magic Quadrant for Enterprise Application Servers, 2Q08"
- "Introducing SaaS-Enabled Application Platforms: Features, Roles and Futures"

**Open-Source AD OOA&D Tools**

**Analysis By:** David Norton

**Definition:** Open-source object-oriented analysis and design (OOA&D) tools support system analysis and solution design in the development life cycle. The tools commonly support OOA&D methodologies. They also support the representation of system syntactical and semantic structure, and behavior at a logical and physical level.

**Position and Adoption Speed Justification:** Open-source design tools, such as ArgoUML, StarUML and Eclipse Modeling Project, have focused on standards-based modeling around the Object Management Group's Unified Modeling Language (UML) specification and the associated
SysML for system modeling in the case of the Topcased UML tool. The tools also offer model-to-code automation with varying degrees of sophistication, but do not have the same coverage as commercial tools for generating a user interface (JavaServer Faces, Struts and ASP.NET); a persistence layer (SQL and schemas); or Java Platform, Enterprise Edition (Java EE), Spring or .NET code.

Open-source design tools have tended to lag behind commercial tools for service-oriented architecture support, with the notable exception of NetBeans, which can be considered enterprise-suitable. Whereas commercial tooling is moving its focus to composite design, open source is still more focused on n-Tier architecture. The tools are also missing the necessary collaboration features, such as a repository for modeling sharing and publishing, and communication tools to support distributed model development.

Potentially, a market lead such as IBM could release an open-source, cut-down version of its commercial products. This would develop a community centered on the open-source version, and would increase the likelihood that a more demanding project would upgrade to the commercial version.

Like the commercial offerings, open-source design tools are experimenting with domain-specific modeling support. However, this is not likely to be of production quality for at least three years.

**User Advice:** Open-source design tools are suited for small teams that do not need sophisticated collaboration features, but need a lightweight modeling solution. Given the current maturity level, the tools are best-suited for non-framework-based (Java EE and .NET), low-technical-complexity projects. Users will find good support for OOA&D in the form of UML modeling, with limited design pattern and legacy engineering support. Logical and physical data modeling are best achieved using the persistence constructs within UML, because traditional entity-relationship diagram design is limited.

**Business Impact:** Open-source design tools offer a viable alternative to commercial offerings where a lightweight approach is needed. They have synergy with agile development where less-rigorous and open design tools complement the agile principles. Given that most open-source design tools are based on the Eclipse platform, they are a natural complement for developers that require code visualization with the Eclipse integrated development environment. This helps to improve the overall quality of the design, decrease documentation and make the review process more efficient.

**Benefit Rating:** Low

**Market Penetration:** Less than 1% of target audience

**Maturity:** Emerging

**Sample Vendors:** ArgoUML; Eclipse Foundation; NetBeans; StarUML; Topcased; UMLet

**Recommended Reading:** "Open Source in Modeling Tools, 2008"

"Telelogic Offers a Free UML Modeling Tool"

"Open-Source Modeling Tools Maturing, but Need Time to Reach Full Potential"

**Open-Source Public-Sector Vertical Applications**

**Analysis By:** Andrea Di Maio

**Definition:** Open-source public-sector vertical applications meet specific public-sector requirements and are developed using a community development model. They are usually
(although not exclusively) associated with an open-source license and are meant to benefit a specific community of users on a particular application domain.

**Position and Adoption Speed Justification:** Open-source applications for the public sector exist on a small but growing scale. The impetus for speeding up their development comes from the need to replace legacy applications, combined with the reluctance of being locked in to application vendors. This is leading to numerous initiatives to pool resources across different agencies and jurisdictions. Examples include state and local governments in France (ADULLACT), Sweden (Programverket.org) and Belgium (CommunesPlone) — as well as those at the international level (PloneGov) and in the higher education sector (Sakai and Kuali).

Examples of nascent communities exist at the U.S. DISA or at the EU level with the OSOR project.

**User Advice:** Consider open-source vertical applications (also known as community source applications) when there is a well-identified and sustainable community of government users and developers who all have a clear business case for participation and support. Assess the need for regional customization/maintenance before committing to a community source project.

**Business Impact:** Affected areas include all those where there are no mature market solutions available or the available solutions lack the necessary flexibility. While local authorities and academic organizations are the most likely target, there is evidence that also national and federal agencies are considering this as an additional application sourcing option. The main benefit for government organizations is sharing the risk of custom application developments. Engaging a more local vendor ecosystem can produce potential additional benefits from an industrial policy and economic development perspective.

**Benefit Rating:** High

**Market Penetration:** Less than 1% of target audience

**Maturity:** Emerging

**Recommended Reading:** "Governments Move Toward Sounder Policies for Open-Source Software"

"Local Governments in France Move to Open-Source Applications"

"When to Use Custom, Proprietary, Open-Source or Community-Source Software"

**Open-Source Technology for Extreme Transaction Processing**

**Analysis By:** Massimo Pezzini

**Definition:** In traditional industries (such as travel, telecommunications and financial services), and in new and emerging sectors (for example, online gaming and betting, Web commerce, cloud computing and software as a service), organizations are experiencing explosive growth in the number of real-time transactions they must sustain. Therefore, applications supporting these businesses are becoming extremely demanding in terms of performance, latency, 24/7 availability, and manifold elastic (up and down) growth in scalability. The competitive nature of these industries poses the additional challenge of supporting these ultra-high-end requirements on top of low-cost, commodity hardware and modern software architectures based on industry standard, Web-oriented technologies and not, as in the previous generation of transaction-processing (TP) systems, on specialized and dedicated software and proprietary hardware.

The essence of extreme TP (XTP) is the ability to support the most-demanding requirements of modern TP applications on mainstream and low-cost technologies. Hence, users and vendors are
exploiting as much open-source software as possible in their XTP strategies. XTP-specialized vendors also are trying to leverage open-source models as a market-seeding strategy, and open-source versions of key XTP technologies, such as distributed caching platforms and extreme transaction processing platforms (XTPPs), are emerging.

**Position and Adoption Speed Justification:** Despite dramatic improvements during the past 10 years, mainstream TP-oriented platform middleware (such as Java Platform, Enterprise Edition [Java EE] and .NET enterprise application servers) cannot address the most-demanding XTP requirements. Therefore, some leading-edge and technically astute users have assembled custom software platforms to support the implementation of pioneering, XTP-style applications by combining mainstream platform middleware (frequently open-source Java EE servers), specialized technologies (such as distributed caching platforms) and open-source technologies (such as the Spring Framework, OSGi Framework and Hibernate). These "custom" platforms often run on top of large clusters of Linux boxes.

Several XTP-focused software companies proposed early examples of XTPPs, specifically designed to support XTP-style applications on top of commodity hardware. Many of these vendors have massively adopted open-source technologies in their products. This way, they aim at reducing their development costs, accelerating time to market, enabling seamless integration with third-party products and technologies, and reducing users' and independent software vendors' (ISVs') learning curves for their platforms. For example, Spring technology and open-source implementations of OSGi Alliance standards play critical roles in many XTPPs, including those under development by large software vendors. To attract users, ISVs and system integrator investments, some XTP pure-play vendors increasingly leverage open-source-style forms of delivery for parts of their platforms, along with free-of-charge distribution of software and other market-seeding techniques. Examples of XTP vendors that are proposing open-source versions (albeit still limited in adoption) of their products include Red Hat (JBoss/Mobicents, a Java application programming interface for Integrated Networks Service Level and Execution Environment-based [JSLEE] XTPP and JBoss Cache), Paremus (Newton Project, an open-source version of the company's Infiniflow XTPP) and Terracotta (Terracotta ES, an open-source, distributed-caching platform). Moreover, many open-source, distributed-caching platforms (JBoss Cache, memcached, SwarmCache, OSCache, ehcache and others) are available from various sources.

The adoption of open-source software in XTP will continue to escalate. Growing competition — including from application infrastructure behemoths — will force pure-play XTP vendors to accelerate delivery of new capabilities and to support emerging standards. Therefore, they will be forced to increasingly integrate open-source software in their products to keep pace with technology evolution. Megavendors needing to catch up rapidly will take advantage of readily available open-source components to develop their XTPPs. Integration of open-source software in XTPP products rapidly will bring to market advanced platforms able to integrate with other software and support established, popular and easy-to-learn programming models, such as Spring. This will favor adoption by mainstream users and will trigger a process of de facto standardization of the fundamental set of features, protocols and programming interfaces of a typical XTPP. Large application infrastructure vendors (in particular, IBM and Oracle) aiming at expanding XTP technology adoption will favor such a standardization process. Therefore, XTPP specialists will converge toward support for the same standards and open-source foundations incorporated in large vendors' products to reduce users' and third parties' natural reluctance to adopt software from small, unknown suppliers. More vendors will deliver functionally limited open-source versions of their XTPPs to win greater industry support as companies like Red Hat, Paremus and Terracotta already are doing.

**User Advice:** Users should recommend that their XTP vendors embrace open-source software as much as is practical. This will drive the XTPP market toward growing standardization (followed
by consolidation) through an open-source-based, semispontaneous process. However, users should bear in mind that XTP requires advanced technology that is unlikely to be commoditized in the near future. Therefore, XTPPs will be based extensively on open-source foundations and will incorporate multiple open-source components, although XTPP products usually will not be entirely open-sourced. XTPPs will continue to be acquired primarily through traditional delivery models during the next five years and until a comprehensive set of XTP standards is established (a precondition for the emergence of viable open-source XTPPs).

In a market still addressed primarily by small vendors of uncertain viability, users should favor adoption of XTP products based on open-source technologies, to mitigate technology and vendor risks. Leading-edge users strategically dependent on XTP for their business operations should invest in active involvement in key XTP-related, open-source initiatives to influence their evolution and acquire relevant skills early in the technology life cycle.

**Business Impact:** The adoption of open-source, building-block technologies will make it possible for vendors to rapidly deliver XTP-oriented platforms and to augment the appeal of their products. Moreover, vendors will exploit open-source-style delivery models to seed the market and to reduce entry costs for users and ISVs. This will enable mainstream users to implement a class of applications that, thus far, only the most deep-pocketed and technically savvy organizations could afford due to the need to procure expensive, complex and proprietary hardware and platform middleware. Market entry barriers for even the most transactionally demanding business models (for example, cloud computing, software as a service, travel reservations or e-payment processing) will be reduced further, enabling users to dramatically lower the costs of their TP systems (thus significantly improving margins for their companies), and will enable new and creative business models.

However, XTP will impose new programming models (for example, natively supported event processing) that will be only partially compatible with traditional Java EE and .NET, new best practices for software architecture, and new management and administration burdens on the IT environments. Typically, this will be in addition to the established system environments that XTP gradually will complement and possibly replace. Thus, IT departments must build and maintain, in-house, the new highly specialized skills (for example, distributed cache administrators) or must delegate highly demanding XTP projects to specialized service providers.

**Benefit Rating:** High

**Market Penetration:** Less than 1% of target audience

**Maturity:** Emerging

**Sample Vendors:** GigaSpaces Technologies; Majitek; Paremus; Red Hat (JBoss); SpringSource

**Recommended Reading:**
"The Challenges of Extreme Transaction Processing in a World of Services and Events"
"The Birth of the Extreme Transaction-Processing Platform: Enabling Service-Oriented Architecture, Events and More"
"Distributed Caching Platforms Are Enabling Technology for Twenty-First Century Computing"
"Predicts 2009: Platforms and Integration Middleware Move Into the Cloud"
"Open Source in Extreme Transaction Processing, 2008"

**OSGi for Application Infrastructure Middleware**

**Analysis By:** Massimo Pezzini
Definition: The OSGi Alliance (originally, OSGi meant Open System Gateway Initiative, but now this characterization has been abandoned) was created in 1998 as an industry consortium whose initial goal was to implement a simple way to dynamically deploy and undeploy Java software on constrained, small-footprint, embedded systems. The OSGi Alliance publishes a Java dynamic component system standard enabling system and application software to be constructed of smaller components that can be dynamically put together in many different configurations, determined by the conditions found in real-life environments.

An OSGi-compliant container provides visibility rules, dependency management and versioning of OSGi modules (called "bundles") that can be installed, started, stopped, updated and uninstalled dynamically at runtime. Therefore, OSGi enables a "service-oriented architecture (SOA) inside" approach to system software by making it possible to install, start, stop and replace software components with zero downtime, including the ability to add and remove functionalities "on demand."

The OSGi Alliance develops free and royalty-free specifications through a process that has some similarities (but also differences in the way intellectual property is managed) to the Java Community Process (JCP) process, and also runs a compliance and certification program to ensure compatibility and interoperability across alternative implementations of the OSGi specifications. When a new version of the specification is to be published, the Alliance makes sure there is at least a reference implementation available from some source.

Position and Adoption Speed Justification: The OSGi specifications have been implemented by at least 12 different organizations (including the Apache Software Foundation, the Eclipse Foundation, Hitachi, IBM, Makewave, ProSyst, Siemens and others), each providing OSGi-compliant products. The most popular and widely adopted OSGi-compliant products are the open-source Apache Felix, Eclipse Equinox (the OSGi reference implementation) and Makewave Knopflerfish. These OSGi-compliant products are used primarily by independent software vendors (ISVs) to support development of modularized software products and, more rarely, by user organizations to develop custom applications. OSGi technology has been used extensively in products from IBM, Nokia, Samsung, Siemens and other providers of home automation and telematic solutions. User organizations today leverage OSGi indirectly, through OSGi-enabled packaged software. As an example, many Java developers daily make use of the OSGi-enabled software in the form of the Eclipse Java integrated development environment.

This trend will continue. Over the past three years, a variety of middleware vendors — including mainstream players like IBM, Oracle, Progress, SpringSource, Software AG, Sun Microsystems and Tibco Software, as well as XTP specialists like GigaSpaces Technologies and Paremus — have openly endorsed OSGi to enable modular, dynamic and optimized SOA-inside deployment of their application infrastructure platforms. Consequently, over the past 12 to 18 months, many OSGi-enabled products hit the market (in many of these products, the OSGi infrastructure is only used internally and is not made available to users).

To increase the appeal for user organizations, the OSGi Alliance will release in the course of 2009 updated specifications (4.2), providing stronger security; support for distributed multi-Java Virtual Machine (JVM) bundle deployments (currently, OSGi supports bundle deployments on a single JVM environment); and a Spring Framework-derived component model (the Blueprint Service, a minor variation of the Spring Dynamic Modules [DM] technology) to make OSGi more appealing for mainstream enterprise developers. With the same aim, OSGi is also working on a Java EE modularization framework specification, likely to be published in late 2009 or early 2010. This will allow OSGi applications to take advantage of Java EE services like Java Database Connectivity (JDBC), Java Persistence Architecture (JPA), Java Transaction Architecture (JTA), Java Management Extensions (JMX) and others.
Adoption of the OSGi-compliant technology will continue to grow as it is increasingly leveraged by platform and integration middleware vendors that will exploit its dynamic modularity to enable optimized and dynamic life cycle management (deploy, update, start, stop and underplay) of system components required to support specific scenarios. Vendors will benefit from this opportunity via greater reuse of proprietary, third-party and open-source technology, and by packaging products in the form of a granular set of more-consumable and easier-to-maintain functionalities than large monolithic platforms.

In particular, the major Java EE vendors have adopted the technology to split their Java EE application server products into a series of well-defined, plug-and-play modules that can be recombined into Java EE functional subsets, in line with the notion of Java EE profiles being defined as part of the Java EE 6 specifications.

Nevertheless, during the next three years, user organizations will primarily continue to leverage OSGi-compliant technology indirectly. OSGi-compliant containers will remain prevailingly adopted in the design of system software, and will rarely be explicitly used by user organizations to develop business applications. However, as OSGi 4.2-compliant products become available, some leading-edge users will likely begin its experimental explicit adoption as a distributed container for distributed business applications.

**User Advice:** In their application infrastructure middleware selection processes, users should favor modularized, SOA-inside (whether OSGi-enabled or otherwise) products over functionally equivalent monolithic products, although prevalingly to establish a platform for future benefits. SOA-inside modularity may provide future benefits to users, in reduced software product footprints, plug-and-play extensibility, enhanced maintainability and upgradability via nonstop operations, and greater software product granularity.

However, users should be aware that OSGi has thus far proved beneficial primarily for middleware vendors in improved time to market and reduced maintenance costs via greater reuse of established software. OSGi-enabled application infrastructure middleware products are just now hitting the market, and their theoretical benefits for users still largely need to be demonstrated in real-life business-critical deployments.

Nevertheless, leading-edge organizations should begin to experiment with OSGi to familiarize themselves with a fundamental enabling technology for 21st-century enterprise computing.

**Business Impact:** OSGi-enabled application infrastructure middleware will make it possible for user organizations to reduce their costs by providing a more incremental and granular way to update their middleware products as needed to introduce new functionalities (whether from the same vendor or from partners), by minimizing application infrastructure middleware redeployments for patching and upgrades, and by enabling acquisition of only the functional components they actually require. This will help organizations optimize the cost of maintaining and upgrading their application infrastructures, and will provide a foundation for greater business agility.

**Benefit Rating:** Moderate

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Apache Software Foundation; Eclipse Foundation; GigaSpaces Technologies; IBM; Oracle; Paremus; Progress Software; SpringSource; Sun Microsystems; Tibco Software

**Recommended Reading:** "Key Issues in Platform Middleware, 2009"
Open-Source Data Integration Tools

**Analysis By:** Andreas Bitterer; Ted Friedman

**Definition:** Open-source data integration tools — such as extraction, transformation and loading (ETL), data federation/enterprise information integration, and replication — conform to the open-source model for software development, deployment and support. Open-source data integration tools enable programmers to read, redistribute and modify the source code of the tools.

**Position and Adoption Speed Justification:** Several open-source data integration tools exist (predominantly in the ETL domain). Adoption by IT organizations in large enterprises, relative to that of commercial data integration tools, remains low. The limited functionality of the open-source offerings is — compared with the commercial market leaders — the reason for the low adoption rate. For example, open-source data integration tools generally support only standards-based connectivity (open database connectivity or Java database connectivity, for example), and lack the ability to interact with packaged applications and nonrelational data sources. In addition, the maturity of metadata management capabilities (specifically, lineage and impact-analysis functionality) and the robustness of development environments are lacking in comparison with commercial tools. However, as general interest in open-source software continues to build, more organizations are asking about the existence of open-source data integration tools. As the deployment of open-source software grows for other infrastructure areas (such as operating systems and database management systems), adoption of open-source data integration tools will increase. In fact, there are already two classes of open-source data integration offerings, those from established vendors with a decent branding and those projects that have near-zero visibility in the market.

**User Advice:** Recognize that the open-source movement has not yet had any significant effect on the data integration market. Although open-source data integration tools exist, they are generally weaker in some functional areas compared with commercial offerings, and they are not being commonly adopted in IT organizations for large-scale, high-complexity scenarios at this point. Because data integration is a complex problem area, a combination of solid technology, services and the availability of skills are key to success. Today, only commercial data integration offerings have a significant combination of these elements. Recognize that the overall cost of ownership of data integration tools includes more than just the purchase price of the software. Use your enterprise's overall open-source strategy and experience with other open-source offerings as a way to judge the potential opportunities and risks of deploying open-source data integration tools in your enterprise.

**Business Impact:** In theory, open-source tools could reduce the cost of implementing data integration processes. This would be applicable to enterprises in all industries because data integration needs are pervasive, as well as to small and midsize enterprises that lack the budgets, infrastructure and skills for more comprehensive data integration tool suite deployments. However, given the state of open-source data integration technology and the general lack of awareness in the market, it is unlikely that, in the near term, this movement will present a challenge for commercial tool vendors. In addition, specialists skilled in open-source data integration tools are not as widely available as those for the commercial counterparts. Often,
organizations deploying open-source data integration products initially adopt a "do-it-yourself" approach and search for help through online communities.

**Benefit Rating:** Moderate

**Market Penetration:** Less than 1% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Apatar; CloverETL; Enhydra Octopus; Jitterbit; Kettle; Pentaho; SeETL; SnapLogic; Talend; XAware

**Recommended Reading:** "Open Source in Data Integration Tools, 2008"

"Magic Quadrant for Data Integration Tools"

"Open-Source Won't Kill the Commercial Data Integration Tools Market Yet"

**Open-Source Service Management Tools**

**Analysis By:** Milind Govekar; Cameron Haight

**Definition:** Open-source service management tools are products offered under several licensing arrangements (GNU General Public License, Apache Software License, etc.), and designed to provide similar ITSM and operations management capabilities as those offered by traditional management providers. These capabilities include performance and availability management, configuration management (including discovery, configuration management databases [CMDBs] and provisioning), help desk, event management and service-level management. The qualitative (feature richness) and quantitative (feature breadth) attributes of these products have continued to improve; so much so that we are observing more companies looking to these products as potential substitutes of those from the larger enterprise management vendors (this discussion is primarily related to those more integrated open-source products from vendors like GroundWork Open Source, SpringSource-Hyperic and Zenoss). Accelerating this trend is the fact that these offerings are usually substantially less expensive than proprietary alternatives — a key benefit in difficult economic times. While they are still primarily marketed outside of the Global 2000, these tools are gaining interest from larger corporations as well.

**Position and Adoption Speed Justification:** Open-source service management products have been available for several years, supporting a variety of IT requirements. Increasingly, many organizations are beginning to use the tools in mission-critical environments because their functionality continues to improve, and standardized support and maintenance contracts have become more widely available. The pressure to reduce IT costs is also prompting increasing interest in these products.

**User Advice:** Understand not only the potential feature limitations (if any) of these products, but also the open-source licenses under which they may be offered. Make sure to assess the supportability and maintenance provisions, especially for products being offered as "projects" or via noncommercial concerns. You may also need to plan for consulting services to address any feature limitations requiring remedy.

**Business Impact:** Open-source service management products can, in many cases, dramatically reduce your IT service and operations management spending in license revenue and associated maintenance.

**Benefit Rating:** High

**Market Penetration:** 5% to 20% of target audience
**Maturity:** Early mainstream

**Sample Vendors:** GroundWork Open Source; Nagios; SpringSource; Zenoss

**Open-Source Virtualization Platforms**

**Analysis By:** Brian Gammage; Monica Basso

**Definition:** Open-source virtualization platforms are layers of virtualization software that run on top of an operating system (OS) or directly on hardware to define multiple virtual machines (VMs). These VMs can be full-user environments or software appliances. The open-source community led the development of a number of virtualization platforms, most aimed at supporting virtualized Linux instances on servers, PCs or smartphones.

**Position and Adoption Speed Justification:** Although developments such as Xen, OKL4, OpenVZ, KVM and rHype have garnered significant market interest, concerns regarding support from commercial OS and application vendors have limited market adoption so far. Xen, KVM and OpenVZ are seeing some deployment on servers, but mainly through packaging with commercial virtualization bundles and/or Linux distributions: Xen has been adopted by Virtual Iron, Oracle, Sun Microsystems, Red Hat and Novel, while KVM shipped with the main Linux kernel from version 2.6.20 and is offered as an external module for commercial distributions. Red Hat's strategic direction for future versions of Red Hat Enterprise Linux is KVM, but it will continue to support Xen. Citrix's distribution of Xen is also expected to be offered with Intel's vPro and CentrinoPro PC platforms beginning in early 2010. OKL4 has achieved some success on commercially available smartphones from Toshiba and HTC, supporting Linux and the Windows Mobile OS.

The leading open-source virtualization platforms continue to be supported by a broad development community, but the core teams that act as their custodians are increasingly being absorbed by organizations that deliver commercial distributions: The leaders of the Xen community work for Citrix (since their acquisition of XenSource in 2007), and the leaders of the KVM community work for Red Hat (following its purchase of Qumranet). Commercial packaging of open-source virtualization platforms will continue to gain momentum, and the need for interoperability will drive mainstream market adoption. However, in most cases, open-source virtualization infrastructure components will be nearly invisible to the IT organization. Open-source virtualization platforms delivered as part of commercial packages will not be "open" for further development.

**User Advice:** Xen, OKL4, OpenVZ and KVM are all viable for deployment and use, in the context of a broader virtualization solution. Of particular importance on servers and PCs is the instrumentation or control interface for managing the VMs, which will mainly be provided by a commercial organization or an independent software vendor. Look for interoperability with other virtualization technologies and the emergence of broad industry support for hypervisors and logical partitioning. In all deployments, the introduction of virtualization software, whether hardware to OS or OS to application, will lead to some degradation of performance because of the additional abstraction processes involved.

**Business Impact:** Open-source virtualization platforms will play a key role in driving embedded hypervisors into computing platforms, making them "free" elements (with associated costs for implementation and support) for many new deployments beginning in 2010.

**Benefit Rating:** High

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Adolescent
**Sample Vendors:** Citrix; IBM; Open Kernel Labs; Red Hat

**PC Linux for Consumers (Mature Markets)**

**Analysis By:** Annette Jump; Michael Silver

**Definition:** Linux is distributed under the GNU Not Unix software license — that is, the operating system (OS) and its source code may be available for free. The term "consumers" refers to home users who employ the desktop for personal productivity, education and entertainment.

**Position and Adoption Speed Justification:** There is relatively little interest in Linux-based PCs in regions with large Windows installed bases. Apple with Mac OS X has been making more progress among consumers in mature markets as an alternative desktop scenario. Lack of compatibility with iPod devices and iTunes is an obstacle to providing the complete consumer entertainment experience. However, mini-notebook success and improved user interfaces helped to increase awareness of Linux in mature markets in the last 12 months. Despite the originally dominant position of Linux on mini-notebooks, it has significantly lost its position on those devices since Microsoft made Windows XP available for them in March 2008. In 1Q09, only 15% to 25% of consumer mini-notebooks were sold with Linux OS in mature markets. Despite some vendor push for Linux on mini-notebooks, consumers prefer Windows, for now, because of its familiarity, wider availability and compatibility of applications.

There might be another opportunity for Linux to increase its share on mini-notebooks after the Windows 7 launch. While the standard Windows 7 Home version should run well on mini-notebooks, its cost will be a significant percentage of the total price of the PC and may encounter some resistance. However, Microsoft will release a version of Windows 7 (Starter edition) at a lower cost to OEMs to compete with Linux. But Windows 7 Starter will have certain limitations (you can run only three programs at once) that will make it less attractive for consumers in mature markets, with exception of the education segment and for some children's mini-notebooks, which are used only as a learning tool for few applications.

While the majority of education mini-notebooks, such as XO devices from One Laptop per Child and Classmate PC, are sold in emerging markets, there is some interest in mature markets (for example, the current project in Portugal with Classmate PCs and small pilots in Germany, the U.S. and the U.K.). This might slightly increase user acceptance of Linux in mature markets.

**User Advice:** Linux desktops are likely to fulfill the needs of consumers with basic computing requirements, limited need for Windows compatibility and limited budgets. However, for most users, there is little perceived need for an alternative to Windows, and users who require support for specific applications, help from friends and family, or mainstream gaming or entertainment support will continue to consider Windows or Mac OS.

**Business Impact:** There is limited availability of applications for general use. Migration efforts, lack of support for Windows applications and a lack of skills are deterrents to wider acceptance; however, for mini-notebooks, Linux has the opportunity to expand its position in the market.

**Benefit Rating:** Low

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Emerging

**Sample Vendors:** Canonical; Debian; Linspire; Novell; Red Hat; Ubuntu; Xandros

**Open-Source Content Management**

**Analysis By:** Mick MacComascaigh; Mark Gilbert
**Definition:** The "open-source software (OSS) model" describes a set of characteristics and properties for developing, delivering and supporting software. Open source is licensed software in which the source code is made available to users so that they are enabled with the freedom to modify it for their own purposes and, with very limited restrictions, redistribute original and derived works as they see fit. "Open-source content management" pertains to the application of this model primarily to document management, workflow, records management, imaging, Web content management and document-centric collaboration.

**Position and Adoption Speed Justification:** An expanding group of vendors is focusing on open-source-based enterprise content management (ECM) technologies. Open source, while attractive, still remains a relatively immature area for ECM. Such tools are, and will remain, primarily of interest to enterprises with the technical skills and resources to adapt the products to fit their requirements, and to counter the risk associated with the perceived lack of a trusted entity behind the product. The number of possibilities on the market is also diffusing what would otherwise be a focused, concentrated development of a particular platform. Interest in open-source ECM is increasing, but only low to modest levels of adoption have been observed in the market. A market driver for this interest is certainly the cost savings for the licenses, or having to pay only maintenance costs, particularly given the current economic climate. However, uncertainty over the necessary quantity and availability of the required services to complete a program of work has given rise to the limited adoption.

**User Advice:** Technically oriented users who are interested in taking on a project will be attracted. The majority of users will continue to turn to traditional offerings. The adoption of open-source ECM will depend in part on the culture of your organization with regard to change, because there are likely to be shorter periods between required updates. Therefore, this replacement of the more-traditional, longer software life cycles will require a very different strategy when it comes to change management. Both intra- and inter-organizational collaboration are recommended and required for best results to ensure optimal reuse and sharing of components already developed. It is through this mechanism that the additional costs for development can be partially offset. However, organizations should not underestimate the overall costs of achieving a solution based on OSS, which is likely to include professional services and/or in-house costs associated with integration, deployment, support and management of OSS. When calculating the total cost of ownership as part of their selection process, organizations should also consider their intended deployment road map and compare the costs accordingly.

**Business Impact:** Open-source content management has generated modest interest among buyers overall, particularly in the current economic climate. However, the perceived uncertainties surrounding the model in terms of support and code quality continue to slow adoption rates. This has given rise to a growing number of companies whose business model is based on providing support and services for a specific open-source content management offering. Most open-source content management tools are primarily of interest to enterprises with the technical skills and resources to adapt the products to fit their requirements.

**Benefit Rating:** Moderate

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Alfresco; Bricolage; CorraTech; Drupal; eZ Systems; Magnolia; MMBase; Nuxeo; OpenCms; Plone; Squiz; Typo3; Xaraya

**Recommended Reading:** "Decision Framework: Use These Criteria to Determine Whether Open Source Should Be a Part of Your ECM Strategy"
Open-Source Horizontal Portals

Analysis By: David Gootzit

Definition: Open-source horizontal portals are built using open-source portal frameworks.

Position and Adoption Speed Justification: Global 1000 companies have yet to implement open-source horizontal portal alternatives in a significant manner, but enterprises worldwide demonstrated increasing interest in open-source alternatives starting in 2008. The category of "open-source portals" is broad and without well-defined shape. Unlike enterprise portal products, the open-source packages in widest use are created by teams without a cohesive understanding of enterprise market requirements. In addition to Linux, The Apache Software Foundation, MySQL and PHP (LAMP)-based packages (such as Drupal and Joomla), enterprises (especially in Europe and Latin America) are increasingly including horizontal portals from commercial open-source vendors, such as JBoss (a division of Red Hat), Liferay, and DotNetNuke, in their technology evaluations. Most of the track record is around LAMP-based portals, but interest in Java-based open-source portal options, as well as those based on .NET is growing. uPortal was developed specifically for the higher education space by a consortium, and has seen significant use in that area.

For the average Global 1000 company, open-source portal frameworks do not always offer compelling advantages in terms of cost or functionality for deployment at the enterprise level. Also, there are significant gaps in open-source software (OSS) portal functionality when compared with the consensus features found in most commercial portal packages and those that are used in many enterprise portal deployment patterns. Some confusion remains in terms of open-source initiatives that use the term "portal," but are more accurately described as Web content-management tools or community publishing tools. Certainly, some functional overlap exists between Web content management and horizontal portals, but the confusion in this case is due to naming choices applying the term "portal" inappropriately.

Although open-source alternatives for application servers have become attractive, open-source portals vary in their focus (content management vs. a horizontal portal framework). Generation 3, Generation 4 and Generation 5 functional attributes are limited in most available open-source portals, although some open-source portal frameworks seek to differentiate themselves in areas such as social networking. The technology in Java-based horizontal OSS portals is maturing, and vendor-independent portal standards, such as JSR-286, are reducing concerns about vendor lock-in. Some open-source horizontal portal alternatives are also leveraging consumer Web widgets to augment their portlet catalogues.

User Advice: Open-source portal alternatives that truly meet the horizontal portal criteria should be considered in enterprise portal evaluations. Users should especially consider open-source portal alternatives for content-centric intranets or content-centric, externally facing Web presences that don't require e-commerce functions. While functional gaps frequently remain between many open-source portal alternatives and commercial options, enterprises facing constrained capital budgets should examine them to determine if they offer "good enough" functionality for their project types.

Business Impact: "Free" software for use by enterprises looking for complete customization of their portal frameworks is available and reduces or eliminates the initial license acquisition costs associated with portals. Commercial open-source horizontal portal options garnered more interest in enterprise settings in 2008 due to cost-cutting pressures.

Benefit Rating: High
Market Penetration: 1% to 5% of target audience

Maturity: Adolescent

Sample Vendors: DotNetNuke; Drupal; Joomla; Liferay; PHP-Nuke; Plone; Red Hat (JBoss); uPortal

Open-Source AD Testing Tools

Analysis By: Thomas Murphy

Definition: Open-source tools and frameworks for performing various types of application testing (for example, unit, functional, regression, load and stress testing), as well as tools to manage test plans, cases and defects, are critical to organizations being able to deliver reliable, cost-effective solutions.

Position and Adoption Speed Justification: Open-source tools for testing and test management tasks have been available for years with varied amounts of adoption. The primary adoption at this point has been by developers that are looking for focused tools for specific problems that have the ability to do the work to integrate these tools. The broadest adoption has been for unit testing (that is, JUnit, NUnit, etc.) and for defect tracking (for example, Bugzilla). Open source is also filling gaps where commercial tools haven't appeared, such as testing for Ruby. The strength of the open-source Eclipse integrated development environment (IDE) as a foundation for tool integration is helping drive faster maturity. Open, extensible browser platforms such as Mozilla are also providing a foundation for direct integration of functional testing tools into the Web browser, thus simplifying use. Open-source solutions are also benefiting from being used in hosted solutions. The adoption of open source by service providers that have technical skills and that want to control license costs also helps these offerings mature. New Web platform providers (such as Google) have a vested interest in building platform testing tools, and Google is the key committer in the Selenium testing framework.

As organizations are pressed for new technology coverage, tighter budgets, and uneven success with commercial tools, we believe that open-source options will continue to grow in adoption. We are also seeing offshore testing providers experiment with open source, which provides a nice combination of tools, support and service. This won't have any large immediate impact on the commercial tools, but, over time, it will create a foundation for erosion.

User Advice:

- Users should look for areas of opportunity, but keep expectations in check.
- Open-source testing tools are most effective for testing teams with programming skills. These solutions tend to require the ability to write code to create tests.
- Unit testing and defect tracking are the most mature.

Business Impact:

- Open-source testing tools can drive quality sooner in the life cycle, reducing overall development costs while keeping development tool costs low.
- Organizations seeking improved return on investment from service-oriented architecture must improve the level of quality.
- Open source helps fill in the gaps in the life cycle, but organizations must consider how they integrate to an overall application life cycle management (ALM) solution.
The success of the Eclipse IDE and Mozilla browser provides common integration and use models for these tools.

**Benefit Rating:** High

**Market Penetration:** 5% to 20% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Google; Mozilla Foundation; SpikeSource

**Recommended Reading:** "Open Source in Software-Testing Tools, 2008"

### Open-Source Application Life Cycle Management Tools

**Analysis By:** James Duggan

**Definition:** Application life cycle management (ALM) tools provide the facility for common process workflow and consolidation of planning, management, measurement and reporting of work item activities in the development process. The tools roll up data previously trapped in point tools (IE, Subversion, Git, CruiseControl w/ANT for CI, etc.) and enable cross-activity, cross-project and cross-tool rollups for more-effective control and delivery of development projects.

Common platforms, such as Visual Studio (although not open source itself) and Eclipse, simplify the market so that the backbone of workflow (and more) may eventually come from an open-source software (OSS) provider. However, most clients will need to include some commercial offerings in their ALM strategies.

**Position and Adoption Speed Justification:** Open-source offerings such as Subversion present good underlying change and configuration facilities, but the workflow and cross-tool facilities are still emerging. Many of the issues are the province of larger teams or more-complex organizations, and will be slower to be incorporated in open-source tools until the members of the open-source communities gain more familiarity with the needed functions and features. Another factor is that the vendors that initially supported Eclipse and similar open-source projects are now trying to protect revenue from further up the stack. Some initiatives, such as ALF within Eclipse, have lost momentum, or are subject to competing initiatives (such as Mylin versus Jazz).

Organizations that want higher-end functionality, and are willing to use an OSS ALM tool, can choose from among a number of proprietary solutions, such as CollabNet and Polarion, that build on top of open source to cover the higher-end function. In other words, it is possible to get high-end capabilities without having to commit to proprietary solutions for the entire product category.

**User Advice:** Larger teams or groups of teams can leverage open-source versioning and configuration tools, but must be cautious of the broader workflow and collaboration facilities. Open-source tools tend to be point solutions, not integrated suites. In the short run, inroads will be made by OSS solutions in organizations with relatively simple structures that don't need suites. However, in the long run, integration will be key. Expect to do substantial integration or to procure some commercial components to implement ALM for the next couple of years. Commercial offerings based on open-source stacks that have already done these integrations will be offered as products and services, and will offer a less-risky way to explore these offerings.

**Business Impact:** Enhanced management and visibility will improve project control and delivery on business expectations. Despite increasing varieties of delivery mechanisms, application management costs will remain moderate.

**Benefit Rating:** Moderate
**Market Penetration:** 5% to 20% of target audience

**Maturity:** Emerging

**Sample Vendors:** CollabNet; Eclipse Foundation; Polarion

**Recommended Reading:** "Open Source in Development Tools, 2008"

**PC Linux for Consumers (Emerging Markets)**

**Analysis By:** Annette Jump; Michael Silver

**Definition:** Linux is distributed under the GNU Not Unix software license — that is, the operating system (OS) and its source code may be available for free. "Consumers" refer to home users who use the PC for personal productivity, education and entertainment.

**Position and Adoption Speed Justification:** In emerging markets, there are two main reasons for Linux adoption or consideration on consumer PCs. The primary reason is the low purchase cost for Linux, and the secondary reason is the desire of some governments to reduce their dependency on Microsoft products or to support their local software providers and standards. Several governments have launched projects or pilots using Linux to bridge the digital divide, but Linux still has a niche position for consumers in emerging markets due to the greater familiarity of Windows OSs and the availability of pirated versions of Windows at a very low price. Piracy continues to increase share for Windows, as consumers often replace the Linux OS with pirated copies of Windows in the first days after PC purchases. Microsoft has been somewhat successful with Windows XP Starter Edition, which it delivered in response to the Linux desktop threat in emerging markets. Piracy still occurs, but Microsoft is focusing on reducing it with its Windows Genuine Advantage program by providing some additional benefits with a legal OS. If Microsoft succeeds and makes Windows more attractive from a price point of view, then Linux might become a less attractive option.

Education mini-notebooks, such as XO devices from One Laptop per Child and Classmate PC, as well as the growing adoption of consumer mini-notebooks, will increase user knowledge of Linux in emerging markets. Despite the originally dominant position of Linux on mini-notebooks, it has significantly lost its position on those devices since Microsoft made Windows XP available for mini-notebooks in March 2008. In 1Q09, only 25% to 35% of consumer mini-notebooks were sold with the Linux OS in emerging markets. Despite some vendor push for Linux on mini-notebooks, consumers for now prefer Windows due to its familiarity, wider availability, compatibility with peripherals and compatibility of applications.

There might be another opportunity for Linux to increase its share on mini-notebooks after the Windows 7 launch, because the standard Windows 7 Home version is targeted at mini-notebooks, too, and has a higher price tag. There will be also a less expensive Windows 7 Starter version available, but it has certain limitations that might not make it attractive for most consumers, with exception of the education segment and for some children's mini-notebooks, used only as a learning tool for few applications.

**User Advice:** The majority of users will need training for Linux because they may lack familiarity with the OS. The availability of consumer applications could be limited, so consider this when requirements are very basic and acquisition costs are a paramount issue.

**Business Impact:** Migration and compatibility are less of an issue in areas where PC penetration is low. Lack of skill remains a deterrent to wider acceptance because many people still have Windows. Cost is key, so if Linux is capable and less expensive than Windows, consumers will try it on both standard client PCs and mini-notebooks. Piracy of Windows continues to be a major barrier to adoption.
**Benefit Rating:** Moderate

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Emerging

**Sample Vendors:** Canonical; Red Flag Software; Red Hat; Ubuntu

### Open-Source Business Intelligence Tools

**Analysis By:** Andreas Bitterer

**Definition:** Open-source business intelligence (BI) tools are BI technologies and application development components that are subject to open-source licensing terms. Open-source BI platforms follow the same BI paradigms as their commercial counterparts, but their vendors mainly generate revenue through service subscriptions. In addition, some vendors provide limited-function open-source versions of their software packages, leaving their full-function commercial package to generate revenue.

**Position and Adoption Speed Justification:** Interest in open-source BI technology is high as organizations look for alternatives to higher-priced commercially available products. Although this technology is being adopted by software vendors developing their own applications for resale, adoption by enterprises as replacements for commercial BI platforms remains relatively low and is growing only slowly. This is due to the additional development skills required to match the capabilities available with commercial BI platforms (in relation to security, scalability, administration, end-user self-service and metadata, for example). But improved marketing and positioning efforts by open-source BI vendors are generating more "pull" in the market, so open-source platforms are being used more often in project-based deployments, sometimes as complements to standard commercial BI platforms.

Open-source vendors are recreating the traditional architecture of the commercial vendors that pioneered the BI platform space. They are doing a good job of providing traditional functions such as reporting, ad hoc querying and online analytical processing (OLAP), and they are supporting them well. However, open-source vendors tend to lag behind their commercial counterparts in delivering innovative, emerging capabilities such as interactive visualization, in-memory analytics and search-based BI.

**User Advice:** Potential customers should be aware that, in practice, open-source BI tools generally ought not to mean free software. Customers should always subscribe to fee-based service agreements to guarantee product support, unless the tools are to be used only in nonproduction test environments. They should recognize that whereas the larger vendors have reasonable support structures, some open-source BI projects are supported solely by the open-source community and lack any service-level agreements. In addition, customers should know that the skills required for open-source BI products are generally hard to find, and that many open-source BI projects are defunct. Unless they aim to develop the source code further, they should steer clear of unsupported projects. Potential users must also understand that the initial cost of an open-source BI software platform is only a small portion of its total cost of ownership.

**Business Impact:** Some open-source BI platform vendors have started to round out their technology portfolios and the "BI stack" in order to catch up with their commercial competitors. Although there is still a significant gap in terms of functionality, scalability and usability, open-source BI tools have advanced significantly to become viable alternatives. However, many smaller open-source BI projects largely provide developer components for adding low-volume reporting capabilities to applications, wrapped in open-source development environments such as Eclipse. The skills required to develop, maintain and integrate these technologies can be much greater than many of the more complete commercially available BI platforms.
**Benefit Rating:** Moderate

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Actuate; Jaspersoft; Jedox; Pentaho; SpagoBI

**Recommended Reading:** "Who's Who in Open-Source Business Intelligence"

"Open-Source Business Intelligence: State of the Market"

"Open-Source Won't Kill the Commercial Data Integration Tools Market Yet"

**Open Source for CRM**

**Analysis By:** Michael Maoz

**Definition:** These business applications support marketing, sales and service processes across interaction channels, built in part or entirely with open-source software (OSS) components.

**Position and Adoption Speed Justification:** Open source will play a strong role in building the foundations of CRM applications, but only a minimal role during the next 36 months in CRM as a commercial set of functional applications. Open-source CRM applications have been on the market for more than five years, yet remain confined to small businesses and departments. Software as a service has been a more compelling alternative as companies reduce core IT resources.

**User Advice:** Look to see what type of open-source approach your prospective vendor uses. Most CRM vendors simply relicense commercial open source. Others own the open-source application, but it is not recognized by the Open Source Initiative (OSI; see [http://opensource.org/docs/osd](http://opensource.org/docs/osd) for the definition of open source developed by the OSI), while a third category of CRM vendors has community-owned open source that is OSI-licensed. Look at the track record of the vendor based on reference checks, especially in the areas of:

- Security
- Scalability
- Cost savings
- Overall issues of process integrity
- Integration
- Ability to support process modeling

**Business Impact:** OSS is mainly used in the foundational technology stack. The primary impetus behind the embrace of OSS by the CRM vendor community has been to drive down the cost of development and total cost of ownership. The tools, technologies, languages and platforms are not monolithic packaged solutions, but rather a modular stack of technology layers. Several CRM startups during the past 10 years would have had much longer and more-expensive incubation periods had they not been able to make use of then-emerging open-source components, such as the base-level operating system, Web/HTTP server, relational database or server-side language platform. The OSS movement is as important for the principle it has reinforced and enhanced in the community of users that heavily impact code and functionality design.

**Benefit Rating:** Low
**Market Penetration:** 5% to 20% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Compiere; SplendidCRM; SugarCRM

**Recommended Reading:** "Open Source in the CRM Application Market, 2008"
"Open-Source Software in CRM, ERP and SCM Business Applications, 2008"

## Sliding Into the Trough

### Open-Source Message-Oriented Middleware

**Analysis By:** Roy Schulte

**Definition:** Messaging systems, also called message-oriented middleware (MOM), are communication subsystems that provide intermediated program-to-program communication for intra-application and interapplication (for example, integration) purposes. The intermediary — which may be software libraries bound to the sending and receiving endpoints, or a message server interposed between the message sender and the receiver — decouples the sender and receiver so that no session is created directly between sender and receiver. This arrangement is the basis for messaging's strengths as a communication mechanism. When used in a publish-and-subscribe communication pattern, messaging supports one-to-one, one-to-many, many-to-one or many-to-many message delivery. When used in a point-to-point communication pattern, interactions may be unidirectional or bidirectional (request/reply). Open-source messaging products are sold as separate utilities or bundled into larger software packages, such as in application servers, enterprise service buses or database management systems. The three industry standards used for open-source messaging products are Java Message Service (JMS), Data Distribution Service (DDS) and the Advanced Message Queuing Protocol (AMQP).

**Position and Adoption Speed Justification:** Riding the coattails of the larger open-source movement, the use of open-source messaging is growing faster than the use of private messaging. The three types of open-source messaging demonstrate vastly different levels of maturity. The position of MOM on the open-source Hype Cycle reflects a blend of the three types of open-source messaging, weighted by their current level of use in the market (thus, heavily influenced by JMS, the most widely used type of open-source messaging).

JMS is stable, mature and reasonably complete. The JMS v.1.1 specification was issued in March 2002 and is maintained as part of the Java Platform, Enterprise Edition family of standards. At least six major open-source JMS messaging products are available, and other less-well-known implementations also exist. Millions of copies of open-source JMS messaging subsystems have been distributed, and many thousands of business applications, some mission-critical, are in production.

DDS v.1.2 is a newer messaging standard, the most recent version of which was published by the Object Management Group in January 2007. DDS is aimed at real-time systems that require very low latency, such as those used in aerospace, defense, industrial automation, other embedded systems and, more recently, in financial trading applications. At least four open-source DDS implementations exist, including Object Computing's OpenDDS (formerly TAO DDS) and the newer PrismTech OpenSplice DDS. Open-source DDS is used in dozens of demanding, performance-sensitive applications, and its use is growing. However, private, closed-source DDS implementations are more widely used.

AMQP is an emerging messaging standard that emphasizes interoperability, in contrast with JMS, which addresses the programming interface, and DDS, which has aspects of both. AMQP was
conceived at JPMorgan Chase between 2004 and 2006. iMatrix’s OpenAMQ is a C/C++ reference implementation, and an Apache incubator project, Qpid, has a Java version. Red Hat is an active participant in Qpid and uses AMQP as a core messaging subsystem in its other products. AMQP interest and support surged during 2H08 and 1H09, although production experience with all versions of AMQP is still limited. The spread of cloud computing is expected to accelerate further adoption in AMQP as the demand for reliable messaging and high performance in cloud applications grows.

The overall messaging market is dominated by private, closed-source products, but open-source messaging products are increasingly popular. Gartner expects use to grow from 2009 through 2014. The use of all types of messaging (private and open source) will continue to expand rapidly, because of the fast growth in asynchronous computing, in general, and in event-driven architecture and publish-and-subscribe, in particular. We don’t expect major new entrants in the open-source JMS messaging market because the major vendors that offer open-source products already offer open-source messaging. However, the DDS and AMQP market segments are less mature, and are likely to see additional entrants and increased activity from vendors.

**User Advice:** Companies should use open-source messaging if they are cost-sensitive, comfortable with other open-source products and have strong technical staff. Companies that are risk-averse or that require extensive messaging management and integration features to handle large, distributed applications should consider buying a support contract from an open-source software vendor. Otherwise, community-supported open-source messaging may be sufficient. Companies should avoid open-source messaging for ultra-high-end workloads (that is, more than 2 million messages per day per application), unless extensive testing, simulation, tuning and monitoring are conducted. Companies with all-Windows environments generally should use Microsoft’s Windows Communication Foundation and Microsoft Message Queuing, rather than open-source messaging; similarly, most mainframe-centric projects would work better on IBM’s WebSphere MQ or another IBM messaging subsystem, rather than an open-source messaging. Finally, open-source messaging rarely is desirable in highly heterogeneous environments with many diverse operating systems and applications written in non-Java programming languages, because private, closed-source messaging handles diverse and legacy platforms better.

**Business Impact:** In appropriate circumstances, open-source messaging reduces the overall application life cycle costs. These products have lower software license and support fees than most of their private, closed-source counterparts (except those that are "free" by virtue of being bundled into another product that companies are buying). Application development and support tasks are comparable, but deployment and technical support tasks can sometimes be higher with open-source messaging. The impact of open-source messaging is visible only to those developing and maintaining the applications and infrastructure; end users do not see a direct impact on their work.

**Benefit Rating:** Moderate

**Market Penetration:** 5% to 20% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Apache Software Foundation; Cisco; iMatrix; Object Computing; PrismTech; Progress Software; Rabbit Technologies; Red Hat JBoss

**Recommended Reading:** "Open Source in MOM, 2008"

**Open-Source Enterprise Service Bus**

**Analysis By:** Daniel Sholler; Jess Thompson
**Definition:** Enterprise service bus (ESB) middleware combines support for service-oriented architecture (SOA), Web services and features from several older types of middleware. An ESB implements program-to-program communication and supports Web Services Description Language (WSDL) and Simple Object Access Protocol (SOAP). In addition, ESBs enable complementary mediation functions that generally include service call redirection; transformation; security; and intelligent, content-based routing. ESBs are based on message-oriented middleware (MOM), which enables communication patterns, such as publish-and-subscribe and message queuing for store-and-forward delivery. Open-source ESBs are a relatively new development. Most of the software has been developed in the last five years, and support models have been offered for a shorter period of time. Although they have a similar range of function, compared with commercial ESBs, open-source ESBs often have areas where the functionality is not as complete, or the range of options is not as broad as the commercial middleware offerings, many of which have been available for 15 years or longer.

**Position and Adoption Speed Justification:** While open-source ESB technology has been around for four to five years, it occupied a small niche for much of that time. During the past few years, the introduction of commercial support for this software, coupled with the economic conditions, has created a surge of interest and adoption. Nearly all ESB discussions today include a consideration of open-source alternatives.

Some components of the open-source ESB, such as the SOAP stack and the Java Message Service (JMS) MOM, are proven functionality. The Apache Software Foundation supports several ESB-related projects, and Progress Software, MuleSource and Red Hat (JBOSS) are aggressively marketing their respective open-source ESB offerings. Even with the pending Oracle acquisition of Sun Microsystems, the GlassFish ESB (Sun's open-source ESB) is likely to continue, because it also serves as the Java Business Integration (JBI) reference implementation for the Java Community Process. However, the commercialization of GlassFish ESB, the Java Composite Application Platform Suite (Java CAPS) service bus, faces an uncertain future. Open-source ESBs lag Linux, JBOSS Application Server, MySQL, open-source JMS MOM and other established open-source offerings in maturity by two to three years. We expect them to gain additional ease-of-use features and production experience during the next three years.

The use of open-source ESBs likely will continue to grow well above the growth rates for proprietary solutions. Historically, the assumption has been that enterprise-class systems required the additional features and capabilities found in the more mature commercial ESBs. However, an increasing number of organizations are finding that the newer open-source offerings meet their needs in many situations, or are opting to do without those capabilities to avoid the upfront investments of commercial products. With several commercially supported alternatives available, open-source ESBs have moved from a developer-oriented curiosity to a viable choice for mainstream organizations. Most organizations will not replace their commercial implementations with open source, but, instead, will use it for new investments in ESB technology.

**User Advice:** All development projects involving 20 or more SOA services or event-driven architecture (EDA) types should use some type of open-source or closed-source ESB technology as the core of their SOA infrastructures, rather than using basic Web services stacks or plain MOM. Organizations with normal demands for their ESBs that are willing to use less-proven products should consider open-source ESBs. Organizations requiring high-availability or highly-distributed systems, or have unusual connectivity requirements, are not likely to be satisfied with the current open-source offerings.

Mainstream organizations should use open-source ESBs in conjunction with commercial support offerings. The issues of viability (of the commercial provider and the open-source project) and the quality of the technology and the support are similar for open- and closed-source offerings. Do
not expect that open source confers any significant difference in terms of vendor lock-in, because, at this time, each commercially supported open-source offering is a unique combination of technologies, and switching providers is a task of similar complexity to switching closed-source ESB products.

**Business Impact:** In general, ESBs improve the quality, manageability and scalability of program-to-program communication; facilitate the sharing of SOA services; and make service versioning and changes to SOA interfaces easier to implement. Open-source ESBs can be used for integration by wrapping portions of legacy and purchased non-SOA applications as SOA services, although the availability and sophistication of adapters may be a constraint. Open-source ESBs are more relevant in SOA service domains in which all the components are new and are written in Java, or provide Web services (WS*) style interfaces.

Open-source ESBs can reduce software costs and enhance vendor independence, especially when other open-source components (such as Spring Framework, Tomcat and Hibernate) are used. Open-source ESBs are putting downward pressure on the price of other commercial ESBs in certain business situations, but will have only a modest overall market effect through 2010, because of the widespread penetration, greater richness and functional maturity of commercial ESBs and related integration technologies.

**Benefit Rating:** Moderate

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Apache Software Foundation; MuleSource; Progress Software; Red Hat/JBoss; Sun (SeeBeyond)

**Recommended Reading:**
- "Toolkit: Request for Proposal for ESB Suite Software"
- "Gartner's Reference Architecture for SOA Application Infrastructure"
- "GlassFish Suite of Products Continues Sun Microsystems' Quest to Leverage Open-Source Technologies"
- "Progress Software Continues to Progress: An Update From Progress Software's Analyst Day"
- "Open Source in ESB Suites, 2008"

**Linux on Desktop for Mainstream Business Users**

**Analysis By:** Annette Jump; Michael Silver

**Definition:** Linux is distributed under the GNU's Not Unix (GNU) software license — that is, the operating system and its source code may be available for free. However, vendors may charge for enterprise-level support. The term "mainstream" refers to a structured task or knowledge worker using an office suite, as well as packaged and line-of-business applications.

**Position and Adoption Speed Justification:** Slow economies, the rising percentage of software costs relative to PC hardware costs, shrinking IT department budgets and Microsoft's pricing policies have led organizations to consider Linux on desktops. Organizations generally target all users, rather than choosing specific groups for which Linux desktops would be appropriate. This adds cost and risk to the projects, makes successful completion less likely and decreases the likelihood of showing return on investment (ROI). Most of the major projects that have generated a great deal of hype are not yet complete. The fact that vendors charge an annual fee for releases of Linux where fixes and support commitments run up to seven years (free versions
generally get security patches for only one year), and most enterprises would have to spend a lot to migrate applications from Windows, has made finding ROI for a migration challenging.

There are few real differences between mature and emerging markets for typical business users. The exceptions are government or education organizations that are considering the possibility of using Linux because of its lower cost or because they don’t want to depend too heavily on Microsoft. At this point, these are mainly just discussions or small pilot programs. However, Microsoft now has a low price for Windows for large education organizations in emerging markets (approximately $3), so the Linux price argument is becoming less relevant.

**User Advice:** Companies should be opportunistic and look for user groups that are good candidates for Linux desktops. These would be easily identifiable groups with a critical mass that depends on relatively few Windows applications. Companies should ensure that there is ROI to cover the cost of supporting an additional operating system in the infrastructure. They should also consider annual support costs that may be required for some commercial builds. In addition, they should not be heavy Microsoft Office users.

The length of time for which a distribution of Linux is supported continues to be shorter than Microsoft’s support for business versions of Windows clients, which could require more-frequent upgrades. As more applications become “OS-neutral,” Linux will become a more-realistic choice; however, most enterprises will require Windows for most of their applications through at least 2011.

**Business Impact:** Applications for general use have only limited availability. Migration and support costs are deterrents to wider acceptance. Some vendors continue to insist on combining the perceived cost-benefits of a switch to open-source office (OSO) products as part of their Linux sales efforts. However, Gartner does not view this as an appropriate approach. A migration to an OSO product does not require moving to Linux on the PC.

**Benefit Rating:** Low

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Emerging

**Sample Vendors:** Canonical; Debian; Novell; Red Hat; Xandros

**Open-Source Social Software**

**Analysis By:** Nikos Drakos

**Definition:** Open-source social software relates to the tools available with an open-source license that encourage, capture and organize open and free-form interaction among employees, customers and partners.

**Position and Adoption Speed Justification:** Some of the more mature open-source social software products are finding their way into enterprises as part of pilot implementations, user-initiated deployments or early-stage tactical deployments. Best-of-breed open-source products that offer specific functionality tend to be at least as mature as their proprietary equivalents, and at least some of them are very visible on the public Internet where they power high-traffic sites. Other, generally applicable open-source benefits such as flexibility and low acquisition costs add to the attraction of open-source social software products.

Although several open-source products enjoy some attention from early adopters, there are important barriers to their broader use and acceptance in mainstream organizations. These barriers include: misunderstandings about open-source software in general and, in particular,
about interpreting the conditions of different licenses; the lack of an accountable and dependable provider; and the extra responsibility for evaluation, deployment, support and general change management. In the case of open-source social software, there is another, perhaps more significant, factor that may limit the relevance of many best-of-breed but stand-alone products — demand for products with more comprehensive functionality is likely to increase, making several best-of-breed products less relevant.

Some of the most prominent examples we have seen being used in organizations include wiki-centric products such as MediaWiki, TWiki and MindTouch, as well as blog-centric products such as WordPress, Movable Type, Apache Roller and Serendipity. In most cases, these products defy clean categorization and are expanding into adjacent areas, either through plug-ins and extensions or through enhancements to the core product. Although these tools can be very good at providing specific functionality, they often frustrate users who are looking for more complete solutions. Moving toward more comprehensive functionality are a number of open-source content management products, portals, community support platforms and Web application frameworks that have been layering social software functionality on top of the core product. Such functionality includes blogs, wikis, profile and membership management, social tagging and friend lists. Drupal, Joomla, Plone, Liferay, eXo Platform and Alfresco exemplify this trend. Open-source products, such as Elgg, that are designed with social networking in mind are still rare.

It is also important to recognize that many proprietary products either bundle or can work with open-source infrastructure services (including directories, application servers, databases and operating systems) or with core components such as search. For example, many vendors include the TinyMCE or FCKeditor and the Lucene search engine as part of their proprietary products, which helps them to offer more complete solutions and reduce their costs.

User Advice: Open-source social software products offer an opportunity for early and low-cost experimentation to test the business value of support for social interactions and to understand its impact and relevance for different business activities. Systematic deployments of open-source social software are still appropriate for organizations with a strong technical capability that includes skills and change management processes that deal with evaluation, ongoing support, maintenance and direct engagement with the developer communities. The critical factors to consider when making decisions about enterprise use of open-source social software are likely to be internal technical capability, commercial support and the availability of suites that combine broader capabilities.

Business Impact: Interest in open-source social software is driven by the same objectives as those behind social software in general — improving connectedness, promoting unplanned collaboration and capturing informal knowledge. One difference, however, is that the low acquisition costs and the "instant gratification" properties of open source are accelerating experimentation and early deployments.

Benefit Rating: Moderate

Market Penetration: 1% to 5% of target audience

Maturity: Adolescent

Sample Vendors: Alfresco; Automattic; Barnraiser; Drupal; eXo Platform; Elgg; Google; Liferay; MediaWiki; MindTouch; Novell; Plone; Serendipity; Six Apart; Socialtext; Twiki.net

Recommended Reading: "Open Source in Social Computing, 2008"

Open-Source DBMS

Analysis By: Donald Feinberg
**Definition:** Open-source database management systems (OSDBMSs) are DBMS engines distributed as open-source software (under an open-source license agreement) and used in applications, including mission-critical applications such as online transaction processing and data warehousing.

**Position and Adoption Speed Justification:** OSDBMSs lag behind proprietary products in terms of high-end capabilities, scalability, high availability and overall maturity. Most are not certified by major third-party software vendors, such as Lawson, Manugistics or SAP. Furthermore, many of the database administration (DBA) tool vendors (for example, BMC Software and CA) do not yet support these DBMS engines. Those that do, such as many business intelligence tool vendors, do so only through the standard Open Database Connectivity (ODBC) interface and not through native interfaces.

Ingres is the most mature of the OSDBMSs, boasting over 30 years of experience in the DBMS market and over 10,000 customers. MySQL is next, and since its acquisition by Sun Microsystems, has been maturing faster toward scalability and reliability. In addition, Sun and EnterpriseDB now have easy-install versions of MySQL and PostgreSQL, respectively, along with tools to manage the installation and environment. As with Ingres, this enables an IT organization to install the OSDBMS normally, without needing to compile code, piece together components and manually manage the environment. There is also increasing interest from third-party software vendors, such as Lawson and SAP, willing to port their applications to these OSDBMS engines. Driven also by the economy and a desire for organizations to reduce costs, use of OSDBMSs is increasing across IT organizations.

The major obstacle for data warehousing using OSDBMSs remains the lack of scalability and high availability needed in the multiple-terabyte sizes currently in use. Today, OSDBMSs are used for small data warehouses up to 500 gigabytes in size, and although we see this changing, we need good proof points or references demonstrating stability and performance along with scalability. In addition, most OSDBMS engines have few data warehouse extensions, including features such as cubes, multidimensional query tables or views, advanced data compression, and specialized data partitioning. Today, most of the larger data warehouse implementations that use OSDBMSs are done with appliances, where the vendor has added intellectual property (additional code) to the OSDBMS. These types of issues will take at least two more years to be implemented and to mature.

**User Advice:** For general, non-mission-critical use, especially supporting Web applications, OSDBMSs can be a good choice, offering a lower total cost of ownership (TCO), a growing number of DBA tools, simplified installation and a solid expertise base. However, only IT organizations with high levels of technical expertise and the willingness to accept risk should consider OSDBMS for mission-critical applications. A major risk is in the level of support from internal resources to manage these OSDBMS implementations, although the DBA tools vendors are beginning to support OSDBMS (for example, Quest Software has five tools now supporting MySQL). For data warehouse implementations, insist on solid references in the database size range, number of users and degree of mixed workload that you expect to have. For all OSDBMS usage, always purchase support options from a vendor in control of the open-source distribution, and do not use an OSDBMS for which support is offered only through the open-source software (OSS) community.

**Business Impact:** Deployments focus on infrastructure supporting Web and portal applications, although there is some use in supporting business applications. In the short term, OSDBMS engines increase the total TCO because of the increase in resources needed to manage them, especially for mission-critical environments. As tools become available to manage the OSDBMSs, there will be an increasing reduction in TCO, primarily from the lack of license fees and lower maintenance cost (see "Cost Optimization with Open-Source DBMSs"). The cost of support
subscriptions will likely be equal or close to that of the proprietary DBMSs, leaving the long-term savings to license and maintenance costs.

**Benefit Rating:** Moderate

**Market Penetration:** 5% to 20% of target audience

**Maturity:** Early mainstream

**Sample Vendors:** EnterpriseDB; Ingres; Oracle; Sun Microsystems

**Recommended Reading:** "The Growing Maturity of Open-Source Database Management Systems"

"Cost Optimization with Open-Source DBMSs"

**Open-Source Office Products**

**Analysis By:** Michael Silver; Annette Jump

**Definition:** Open-source office (OSO) productivity products — such as Sun Microsystems’ StarOffice Software and the OpenOffice.org (OO.o) project that Sun uses as its base — are low-cost office suites that offer much of the functionality of Microsoft Office, as well as moderate compatibility with Microsoft Office at a lower cost.

**Position and Adoption Speed Justification:** Interest in OSO products has been high, but adoption has been modest. One major reason for this disparity is that older versions of Microsoft Office, which continue to work well, have already been bought, paid for and deployed. Migration technologies from Microsoft Office to OSO are improving and maturing. Microsoft’s inclusion of OpenDocument file (ODF) converters in Office 2007 SP2 and International Organization for Standardization (ISO) approval of Office Open XML, based on the format included in Office 2007, have some potential to improve interoperability and OO.o adoption, but interoperability will never be perfect, relegating OO.o to be a solution that cannot satisfy all users.

Organizations continue to investigate OSO for some users to bridge the gap between the number of devices that need licenses and the number of Microsoft Office licenses they already own. Although the economy is forcing some organizations to look at less-expensive alternatives to Microsoft Office, layoffs at many organizations with releases that are still supported and viable mean that fewer organizations have a shortfall of licenses that has driven the investigation of OSO.

**User Advice:** Enterprisewide adoption of OSO is still not be a viable option, because users must first be classified and selected as appropriate. Evaluate business and cost benefits before moving users to OSO products. Some users will be easier to migrate than others. Monitor the evolution of open standards for document formats and the availability of converters.

**Business Impact:** Organizations look at OSO products to save money by using “free” products. However, migration costs are significant, and the compatibility of OSO products is not as good as companies need it to be to serve all their users. In most organizations, at least some users will continue to require Microsoft Office, creating the requirement for a mixed environment, the complexity of which most managers find undesirable.

**Benefit Rating:** Low

**Market Penetration:** 1% to 5% of target audience

**Maturity:** Adolescent
**Sample Vendors:** IBM; Novell; OpenOffice.org; Sun Microsystems

**Recommended Reading:** "Considering Alternatives to Microsoft Office"

**Linux on 16 to 64 Processors**

**Analysis By:** George Weiss; Andrew Butler

**Definition:** This technology involves the ability of the Linux operating system to support servers capable of symmetric multiprocessing (SMP) using as many as 64 processors.

**Position and Adoption Speed Justification:** Linux is no longer untested in production environments on servers using as many as 64 processors. Vendor benchmarks and production installations indicate that Linux can support applications using online transaction processing (OLTP) approaching the performance of Unix on 16- to 64-core SMP servers. Many vendors — including Unisys, SGI, HP, IBM, Sun Microsystems, Fujitsu, Bull and NEC — are investing in Linux-based strategies for business-critical applications. This spans systems with nonuniform memory access (NUMA) architectures, particularly in scientific applications, and covers multiple architectures, including mainframes. For example, Fujitsu has submitted an audited TPC-C benchmark on a 64-core Itanium Primequest of nearly 2.38 million transactions per minute on Red Hat v.4, with an Oracle Database 10g. (We advise users to run their own benchmarks, because different workloads will yield different results.) Considerable progress in scaling occurred through the 2.6 kernel releases (the last such major numbered release of the kernel).

As vendors deploy reliability, availability and serviceability (RAS) functionality at platform, rather than operating system) level, Linux (and Windows) is narrowing the gap between it and Unix, as the leading vendors support Linux on their reduced instruction set computer (RISC) and Itanium platforms, as well as x86 (including blades, farms and SMP servers with more than four sockets). Although improvements are still needed in trace profiling, root cause analysis, higher-performance file systems and performance analysis tools, we expect to see many of these improvements during the coming 12 months. These enhancements will come from contributions by the kernel community, as well as from experienced Unix vendors and chip designers, including staff at Intel and AMD.

In the meantime, next-generation Linux applications will appear from independent software vendors (ISVs) for scalable SMP systems. ISVs are capitalizing on the market opportunity for Linux, even though the major Unix vendors — IBM, HP and Sun Microsystems — will not abandon their Unix developments. As a result, the high end of the market will be the last bastion of Unix and will slowly decline but not disappear in the next five to 10 years. Users will be at different points on their risk curves in deploying Linux on higher-performance SMP servers, and the 16- to 64-core market will remain a minor part of the total server market, less than 10% to 15% during the next 12 months.

**User Advice:** Monitor production installations in this class of scalable systems, and ask vendors to provide itemized checklists comparing Unix system tools and RAS features with those of Linux. Verify that traditional Unix RAS benefits that are now delivered as a platform function are adequately proven.

Recognize that Unix will continue to be a viable operating system through 2014 and beyond, although growth will flatten or decline. Most Linux expansion will continue to come from lower-end, commodity-type servers in the one- to four-way configurations, with the addition of scientific and technical SMP and NUMA systems of eight-ways and beyond.

As on-chip multiple cores are added, some applications may no longer need the scaling of midrange Unix SMP systems (eight-way configurations, for example). Users can consider Linux on RISC and mainframes as viable niches for some applications. This includes large-scale
database management systems serving for OLTP and data warehouse deployments. For most applications on servers in the 16- to 32-way range, Linux is a self-tuning and optimizable operating system that requires little or no intervention by data center engineering.

**Business Impact:** Assuming that the adoption of Linux delivers significant total cost of ownership (TCO) benefits is not necessarily automatic because manageability, availability and uptime support response to service-level agreements should be considered part of TCO. Back-end database management and high-performance applications with low-latency or consolidation requirements have a positive business impact. Users will be more likely to derive benefits if databases and applications can be architected to benefit from horizontal scaling trends, where applicable.

**Benefit Rating:** Moderate

**Market Penetration:** 5% to 20% of target audience

**Maturity:** Adolescent

**Sample Vendors:** Fujitsu; HP; IBM; NEC; SGI; Unisys

**Climbing the Slope**

**Spring**

**Analysis By:** Eric Knipp; Massimo Pezzini

**Definition:**

Spring is an open-source framework for building Java and Java Platform, Enterprise Edition (Java EE) applications, providing a component model and additional features for common problems in Java development. Spring’s value proposition is based on simplifying application development through dependency injection (DI) and aspect-oriented programming (AOP), and on its role as a lightweight container for Java components.

Spring Core includes modules for common development needs, which are broken into fine-grained components for selective adoption. For example, modules for transaction and persistence management that leverage object-relational mapping (ORM) frameworks (such as the Java Persistence API [JPA], Hibernate and EclipseLink, through a common interface) are included, as is a more basic Java Database Connectivity (JDBC) abstraction layer. In addition, Spring Core includes the Spring Model-View-Controller (MVC) Web framework, which helps developers build multitier Web applications, and other components, such as Spring Security, Spring Web Flow and Spring Batch.

Spring Core is container-neutral, so it can be used in conjunction with any Java EE server, such as WebSphere, WebLogic or JBoss, in addition to lighter alternatives like Tomcat or Jetty, Open Services Gateway initiative (OSGi)-based containers like Spring DM Server, or extreme transaction processing (XTP)-oriented platforms from vendors such as GigaSpaces Technologies, Appistry, or Majitek.

Spring technology is also converging with OSGi standards. The new OSGi Blueprint Service is largely inspired by, and compatible with, Spring Dynamic Modules, an OSGi-enabled variant of the Spring Core.

**Position and Adoption Speed Justification:** Because of Enterprise JavaBeans’ (EJB’s) reputation as cumbersome, Spring was quickly adopted by Java practitioners and has been widely used by user organizations, independent software vendors (ISVs) and system integrators.
Spring is supported by every significant Java development environment, including those based on Eclipse. At least 50% of enterprises are using Spring in their Java development.

Since its initial release, Spring has proved very popular and has been downloaded more than six million times. While the broader Java EE market is fairly stable, organizations are increasingly adopting agile techniques, which is driving the examination of dynamic programming tools, like Groovy and Grails, for rapid development. The Grails Web framework integrates Spring, and, as it displaces some EJB-centric Java EE development, it will accelerate its growth in the enterprise.

Spring makes it possible for developers to externalize dependencies, simplify unit testing, or leverage Spring’s flexibility and ability to integrate with EJB, JavaServer Faces (JSF), Struts and other frameworks. Organizations using Spring range from smaller companies that use it in unmanaged applications, to large enterprises that depend on Spring in conjunction with Java EE, to ISVs leveraging the framework in the context of system or application software products. The technology is supported by a large and active community, and Spring skills and know-how are widely available from a variety of sources. Although competing open-source frameworks abound, none has reached the same penetration as Spring or has the same comprehensive set of features.

The strategic relevance of Spring also lies in its container-neutral nature, which positions the technology as a natural “bridge” between the current generation of monolithic, on-premises Java EE-based application servers and the next generation of highly modular, elastically scalable, XTP and cloud-oriented application platforms (XTP platforms and cloud/software-as-a-service-enabled platforms).

Large and powerful vendors, such as Google, IBM, Oracle and Tibco, support Spring in the context of application-platform-as-a-service (APaaS) offerings or innovative forms of platform middleware addressing requirements like event-driven applications or XTP. Spring convergence with OSGi will further favor its adoption by ISVs wishing to improve modularity and dynamic deployment of their products via OSGi.

Although Spring adoption, so far, has been primarily focused on supporting Web applications, its adoption for large-scale transactional deployments is rapidly growing and — thanks to support from XTP-oriented vendors — it is the primary candidate to becoming the reference programming model for certain classes of the next generation of global-class, business-critical, cloud-enabled applications.

**User Advice:** Users and ISVs looking to improve flexibility in their Java application architecture, especially with respect to configuration, should consider adding Spring to their technology portfolios. It can help simplify dependency management to enhance unit testing or test-driven development approaches. Spring also allows for incremental and hybrid adoption because it can be used in concert with other frameworks. Moreover, it enables development of applications that can be relatively easily migrated from Java EE containers, Web containers, XTP platforms or even Java Platform, Standard Edition to using container-specific implementations and, in the future, cloud platforms. Spring adoption is facilitated by the wealth of documentation, training and support available in the industry, and by the ample supply of skills that make its learning curve relatively painless and not excessively expensive.

Adopters should take into account that the a la carte approach of Spring may be confusing for some users. Not all components of the Spring technology are fully integrated with each other (e.g., Spring Batch is not yet integrated with Spring DM Server). For highly stateful applications, EJB 3.0 technology may be a more natural fit than Spring. Even if support from large vendors is growing, Spring is, for the most part, guided by a relatively small company, SpringSource, which may pose long-term sustainability challenges, especially for the less-popular components of the Spring portfolio.
**Business Impact:**

Spring helps organizations rapidly develop across a wide range of application styles: from opportunistically oriented solutions aimed at addressing short-term, transient business opportunities, to business-critical systems meant to support core processes. Its container independence provides user organizations with a range of scalable deployment options (from lightweight containers like Tomcat, to Java EE servers, to XTP platforms and all the way to APaaS) that are capable of flexibly following their evolution as they scale up and down to adapt their operations in accordance with changing business conditions.

**Benefit Rating:** Moderate

**Market Penetration:** 20% to 50% of target audience

**Maturity:** Early mainstream

**Sample Vendors:** GigaSpaces Technologies; Google; IBM; Majitek; Oracle; Paremus; SpringSource; Tibco

**IT Services for Open-Source Software**

**Analysis By:** Bob Igou

**Definition:** IT services for open-source software (OSS) include professional services and product support services. Professional services include consulting and system integration services to design, construct and manage solutions; system integration assistance in the selection and deployment of various OSS components; and environmental assessments to detect and inventory OSS components. Product support services include software testing, validation and maintenance (upgrade and patch) services; software support and hardware platform support.

**Position and Adoption Speed Justification:**

A Gartner survey completed in November 2008 of IT organizations in Canada and the U.S. asked about their use of OSS and the demand for related IT services. By the end of 2009, of the total software portfolio, they expect 46% to be proprietary and 22% to be open source, and the remaining 32% to be internally developed. Asked about their overall use of OSS in their organization, respondents indicated that 37% was used as an alternative to proprietary software, and 23% was used as a replacement for proprietary software.

Respondents in the survey assessed how much of their proprietary and OSS portfolios were mission-critical in nature. They estimated that by the end of 2009, 65% of their proprietary software would be classified as mission-critical, and 49% of their OSS would be mission-critical. Compared with the classification of software installed now, the mission-critical percentage of OSS will increase by 3.5% at the end of 2009.

The same survey shows a significant increase in respondents who are planning to use IT services for OSS in the next budget year. Thirty-three percent of the respondents identified specific IT services they currently procure from an external service provider (ESP) related to OSS. Asked about plans for the next budget year, more than double the number of respondents (71%) identified plans to use specific services in the next budget year. Respondents estimated 83% of the current OSS-related service budget to be used for product support and professional services. Overall, this indicates that the OSS portfolio and demand for related IT services have matured to look more like the characteristics we would expect for a proprietary software portfolio.
Although adoption of open-source software is positioned at more than 50% of the target audience, the development of IT services always lags behind market adoption. For this reason, we expect the Time to Plateau to be in the two- to five-year range.

**User Advice:** The mission-critical nature of OSS is driving an increased use of services from ESPs. Within specific IT organizations, this expense starts small and will grow as the use of OSS expands.

Our advice to users is as follows:

- IT organizations need to begin quantifying their expenditure with ESPs for both open-source and proprietary software categories so they can budget appropriately in future years, as their use of OSS becomes mainstream in their own organizations.

- IT organizations need to expand their governance policies to include OSS, and they will need to review and consider open-source support providers not previously used.

**Business Impact:** The trends in these surveys indicate that a majority of IT organizations are consistently predicting the use of categories of IT services, such as software support and consulting provided by ESPs for OSS. Respondents estimate the proportion of spending on OSS for the next fiscal year to be about 32% of their entire budget for ESPs, and this number is expected to continue to grow. The budget for ESPs includes services such as hardware maintenance and support, software maintenance and support, consulting services, development and integration services, managed services, and IT outsourcing services.

Our positioning of IT services for OSS is based in part on the degree of utilization of OSS software. In regions or countries where utilization of OSS is not as advanced, IT services will also be lagging our positioning in the Hype Cycle.

According to the survey, utilization of open-source in the infrastructure software categories is approximately twice that of application software categories. This trend is consistent with previous surveys.

**Benefit Rating:** Moderate

**Market Penetration:** More than 50% of target audience

**Maturity:** Mature mainstream

**Sample Vendors:** Amentra; Black Duck Software; Capgemini; EDS; Exadel; HP; IBM; Interface21; MySQL; Novell; OpenLogic; Optaros; Oracle; Red Hat; SourceLabs; SpikeSource

**Recommended Reading:**
- "Understanding Gartner's Hype Cycles, 2008"
- "Forgoing Linux/OSS Maintenance as a Cost Optimization Strategy Can Have Negative Consequences"
- "The State of Open Source, 2008"
- "Open-Source Software Impact on IT Services Purchasing Patterns, 2008"
Mission-Critical Workloads on Linux

Analysis By: George Weiss; Donna Scott

Definition: This technology includes all the operations and critical foundations that enable an organization to function 24/7, including all necessary "ecosystems." Linux must be accepted as essential to an organization in every way if it is to accommodate complex, mission-critical workloads, including effective support for databases, recovery, disaster tolerance, system management, service-level agreements and dynamic resource allocation. Additionally, it needs to work well within the conventional skill sets of mainstream IT departments.

Position and Adoption Speed Justification: The market is moving beyond the leading-edge adopters. Many users who are considered mildly conservative are using Linux for increasingly demanding workloads, including database management (for example, DB2, Oracle and MySQL), midtier applications (such as those from SAP), and Web services and e-commerce applications. The reasons are because Linux: (1) scales better (to 64 cores); (2) has solid vendor support in technical services from platform and software vendors, such as IBM, HP, Unisys, Fujitsu, NEC, Oracle and SAP; (3) has improved coordination from the distributors, such as Red Hat, Oracle and Novell; and (4) has a complement of third-party open-source software (OSS) service vendors with mission-critical support (such as OpenLogic and SpikeSource). Additionally, users understand that they must invest time and effort in configuration, performance analysis, testing and deployment, but their skills from Unix have been good preparation.

Significant progress has been made during the past 24 months on the kernel (scalability and reliability), storage and volume management, as well as with larger user populations (in the thousands). Improving performance continues to be demonstrated on platforms from HP Integrity, IBM's System z and System p, Unisys, SGI, Fujitsu, Sun Microsystems and NEC. With the latest distributions of the kernel from Red Hat and Novell, improvements to the kernel scheduling and input/output, as well as cluster file systems (such as OCFS2 and GFS) have enabled Linux to scale in symmetric multiprocessing (SMP) and cluster configurations. Although Linux is primarily hosted on x86 technology, the lack of interest in Linux on traditional SMPs seems to be mostly a factor of costs. Users want the lower entry and incremental costs of x86, and are focused on OSS adoption and portability of these platforms as opposed to RISC/Itanium.

More users are seeking lower-cost hardware by migrating applications from high-end Unix systems to free up capacity on the more-expensive hardware, and reduce the upgrade cycle. At the same time, x86 platforms are moving upscale into multisocket, multicore designs to replace more midrange Unix servers with some of the benefits of SMP designs. The biggest exception to this trend has been IBM's success in selling consolidation of Linux instances on the System z. Approximately 15% of IBM's mainframe MIPS are running Linux in specialty engines known as Integrated Facility for Linux (IFLs). We continue to adjust Linux's position on the Hype Cycle to reflect the technological progress against the market's willingness to create a Linux-ready enterprise from top to bottom.

User Advice: Enterprises should expand the use of Linux to take advantage of commodity hardware and the associated cost savings. If you have well-trained resources, then using Linux in mission-critical and highly available environments is acceptable. Investigate hardware vendor-based support options.
Linux kernel maintainers and distributors have brought the performance and reliability of the kernel to nearly the scale of Unix. Unix benefits in performance mostly by the richer platform vendors, such as HP, IBM and Sun, but Linux can run on RISC and Itanium servers as well.

One of the significant drivers toward mission-critical workload support has been the Oracle Real Application Clusters (RACs) 11g environment for multinode, clustered x86 servers. Oracle provides technical support for its database systems, middleware, applications and its Linux enterprise support. Other helpful factors include Novell's relationship with SAP to ensure support Level 1 through Level 3, with handoff transparency. Dell, HP, IBM and Sun also offer varying degrees of support, including all-level support. Integration and support skills for large applications and database workloads on 64-bit processors and large memory configurations have been available for Linux in production.

Users should continue to track developments by the kernel community on core dump, trace and diagnostic utilities, as well as efforts by Linux distributors and third parties to improve and deliver on highly available, disaster recovery, event management and monitoring tools.

**Business Impact:** The business impact remains high. The primary advantage is the promise of lower total cost of ownership (TCO) and vendor flexibility. In organizations starting out with Linux, these cost savings may be hidden for the first few years as resource skills and competencies are developed. Most organizations report cost advantages after attaining these skills and tools, but complexity will have a bearing on how large the savings are, or whether any savings are gained at all. The greater level of flexibility with a Linux platform from the number of available hardware choices can also contribute to lower TCO in hardware contract negotiations. Finally, larger and more-affordable configurations can be achieved with blades and clusters becoming more scalable in cores and, therefore, acting in a horizontal and vertical capacity mode.

**Benefit Rating:** High

**Market Penetration:** 20% to 50% of target audience

**Maturity:** Mature mainstream

**Sample Vendors:** Canonical; Dell; Fujitsu; HP; IBM; NEC; Novell; Oracle; Red Hat; SAP; SGI; Unisys

**Recommended Reading:**
"Case Study: Community Connect Inc., a Linux/OSS Server Infrastructure Predating Facebook and MySpace"

"Linux Case Study: The U.S. Air Force Goes From Big Unix 'Iron' to x86 Linux in 290 Days"

"Oracle RAC Moved to Mainstream Use"

**Linux on System z**

**Analysis By:** John Phelps; Mike Chuba

**Definition:** Linux on System z is the Linux distribution that has been enabled to run on the IBM mainframe z/Architecture. By using the Integrated Facility for Linux (IFL) specialty engine and the z/Virtual Machine (z/VM) virtualization software, it is able to deliver good performance and throughput to arrange Linux applications and workloads with a competitive total cost of ownership. This includes applications traditionally run on distributed architectures (for example, Web-based HTML pages, e-mail and file/print) and traditionally complex applications, such as database and transaction processing.

**Position and Adoption Speed Justification:** Linux usage on the mainframe continues to grow — in terms of the number of customers, as well as the number of IFLs per customer. The number
of customers using Linux on System z exceeds 1,300, up approximately 25% from two years ago. The number of shipped IFLs exceeds 7,000, up nearly 60% from YE06 as many existing users of Linux on System z expand their use and add additional IFLs. We estimate that Linux "MIPS" represents approximately 15% of the worldwide installed base of IBM MIPS capacity of more than 14 million MIPS. The application mix goes from simple infrastructure type applications, such as Web, e-mail and file/print, to complex applications, such as SAP applications, Cognos business intelligence, and Oracle 10g Real Application Clusters (RAC) database applications. The use of the IBM System z virtualization capabilities of logical partitioning and z/VM allow very large numbers of Linux systems to share a single mainframe system with multiple diverse workloads.

The high availability of hardware (dynamic processor sparing), potential cost savings (high levels of consolidation and reduced software license charges based on fewer processors) and tighter integration with mainframe applications (with the use of the internal network — HiperSockets) are helping to drive growth. One issue is the lack of good information regarding performance that compares Linux on the mainframe with Linux on other platforms. The cost of IFL engines and z/VM will mean that a significant number of Linux instances are required to make a business case for Linux on System z. To counter this, IBM continues to improve the price/performance of Linux on System z with faster IFLs with each new generation of processors (with no increase in price) and to lower the price on smaller systems, with the IFL price reduced from $95,000 list price to $47,500 on the System z10 Business Class (BC). We have begun to see large companies place major mission-critical workloads on IBM System z — on mixed and Linux-only mainframes. Major software vendors have begun to announce support for this environment. The number of Linux applications supported by independent software vendors on the IBM System z topped 2,400 by YE08.

User Advice: Mainframe users should examine the use of Linux on System z for applications that communicate with mainframe applications via TCP/IP. The use of HiperSockets can simplify and secure the network connections (all internal) and can enhance the performance (shortened TCP/IP stack and memory-to-memory move). Where applications exist on mainframes and Linux, and that offer similar functionality, consider the cost savings that might be accrued by offloading the work to less-expensive IFL specialty engines. The IFL will not affect the cost of legacy software, and any work offloaded to the IFL will open up space for legacy applications to grow with no additional costs. Do not consider engineering/scientific workloads for Linux on System z, and be wary of other heavy compute-intensive workloads. (Although the System z10 has enhanced capability for compute-intensive workloads, users should still carefully examine performance before committing to workloads on the System z10.) When looking at implementing Linux on System z versus other platforms, consider the higher hardware availability of System z and possible software price savings due to the fewer number of processors needed and how that may impact software license fees.

Business Impact: There is a consolidation of front-end applications with back-end databases through HiperSockets and high-availability offerings. Linux on System z enables major consolidation of workloads that previously required separate servers, with associated hardware, software and environmental savings in areas where large numbers of Linux systems can be combined. These workloads can be consolidated on System z with little impact on mainframe legacy workloads. The reason is that they run on IFL specialty engines and, therefore, do not use general-purpose processor power and do not affect legacy software charges.

Benefit Rating: High

Market Penetration: 20% to 50% of target audience

Maturity: Early mainstream

Sample Vendors: IBM System z; Novell; Red Hat
Recommended Reading: "How to Decide on a Linux Server Platform"

Open-Source Java EE Application Servers

Analysis By: Yefim Natis; Mark Driver; Massimo Pezzini

Definition: Java Platform, Enterprise Edition (Java EE) application servers are implementations of Java Community Process Java EE specifications (formerly known as J2EE) that have passed the required specification compliance tests. Open-source Java EE application servers are, in addition, developed and maintained by an open-source-style community, and are distributed under an open-source license.

Position and Adoption Speed Justification: Open-source Java EE is a proven alternative to closed-source-vendor Java EE implementations and other application server technologies, such as Microsoft .NET, Zend PHP, Adobe ColdFusion and others. Although only the Red Hat JBoss application server (and the uncertified Java EE subset, Apache Tomcat) are widely used, multiple alternatives are available. There is a significant and steadily growing production use of these products, and the user experience has been positive on the technical quality, pricing and support aspects of most offerings. All the leading Java EE vendors have acknowledged the long-term staying power of open source in their business spaces. IBM offers a complement to the WebSphere application server based on the Apache Geronimo open-source Java EE implementation; Sun Microsystems had replaced its formerly closed-source flagship implementation of Java EE with the open-source GlassFish project technology, although the future of its commercial support is in doubt pending the acquisition of Sun by Oracle.

Bull of France continues to invest in OW2 JOnAS. The leaders, Red Hat JBoss application server and Apache Tomcat, are mature and widely used by mainstream enterprises and independent software vendors alike, while the competing open-source Java EE offerings have not altered the market alignments. Thus, the open-source Java EE market retains its viability and, in fact, experiences steady growth, reflecting the recently increased interest in lower-cost software solutions. The most promising alternative to the Java EE programming model is the open-source Spring Framework, but even Spring is typically layered over a JBoss or another Java EE application server or Tomcat, thus supporting the growth of the overall open-source Java EE market. The leading providers of Java EE implementations embed some open-source components in their closed-source offerings, quietly endorsing the long-term value of the open-source software model and adding to its long-term viability.

User Advice: Mainstream enterprises can assume that the use of open-source Java EE application servers (with either Java EE or the Spring Framework programming model) will remain a common characteristic of the enterprise Java environment, along with other available options, such as closed-source Java EE, Microsoft .NET, PHP and others. Those that currently use open-source Java EE application servers can increase their commitment; those that were avoiding them can begin deployments. Although the most-demanding, high-end applications continue to depend on closed-source, partly vendor-proprietary implementations, most projects can be successfully deployed on one of the leading open-source alternatives.

Business Impact: Some key characteristics of open-source Java EE implementations are attractive to users, and are influencing the strategies of the dominating closed-source Java EE vendors — open-source Java EE product distribution is by low-cost subscription, and the internal design is highly modular and typically microkernel-based ("SOA inside"), enabling incremental deployment and engineering. During the next three years, open-source Java EE implementations will continue to pressure leading software vendors to offer low-cost alternatives to their expensive closed-source products. We expect the vendors to separate their differentiating technology into high-cost extensions or variants to the basic platform offerings, while experimenting with subscription pricing and embedding for the core application platform technology. The costs of
basic platform technology will decrease, but the costs of differentiated high-end and vertical extensions will increase. Although initial technology acquisition is much less expensive with open source, the total cost of ownership may not be lower, depending on business practices and application styles that utilize the platforms.

We do not believe that open source will endanger business viability of traditional closed-source alternatives. The service-oriented-architecture (SOA)-inside option, implemented using the Open Services Gateway Initiative (OSGi), JMX or another standard, is becoming widespread among closed-source Java EE and other platform technologies, improving the flexibility in selling, buying, maintaining and managing software products in enterprise IT. Embedded open-source, community-style development including agile project management methods, and open-source-inspired microkernel-style modularity and subscription pricing, are all direct consequences of the open-source initiatives in the software industry, and all have business consequences in agility, costs and distribution of responsibilities in the software industry.

**Benefit Rating:** Moderate

**Market Penetration:** 20% to 50% of target audience

**Maturity:** Early mainstream

**Sample Vendors:** Apache Software Foundation (Geronimo); OW2 Consortium (JOnAS); Red Hat (JBoss); Sun Microsystems

**Recommended Reading:** "Sun Middleware Under New Management: What to Expect"

"GlassFish Suite of Products Continues Sun Microsystems' Quest to Leverage Open-Source Technologies"

"Open Source in the Application Server Market, 2008"

"Magic Quadrant for Enterprise Application Servers, 2Q08"

"Key Issues for Platform Middleware, 2009"

"The World After BEA Systems"

**PC Linux for Data Entry Workers**

**Analysis By:** Annette Jump; Michael Silver

**Definition:** Linux is distributed under the GNU's Not Unix (GNU) software license — that is, the operating system and its source code are available for free. However, vendors may charge for enterprise-level support. The term "data entry worker" indicates limited function as a replacement for PCs running few applications, appliances or terminals.

**Position and Adoption Speed Justification:** The perception of low cost with added functionality, compared with terminals, makes Linux attractive. A limited need to run Windows applications makes migration easier. Linux continues to be used in these cases, but isn't expanding rapidly.

**User Advice:** Linux could be an appropriate product for data-entry workers, but migration costs, application compatibility and support costs still must be evaluated.

**Business Impact:** Linux can be used on the PC for point-of-sale, kiosk and data entry functions.

**Benefit Rating:** Low
Market Penetration: 1% to 5% of target audience

Maturity: Emerging

Sample Vendors: Linspire; Novell; Red Hat
Appendixes

Figure 3. Hype Cycle for Open-Source Software, 2008
<table>
<thead>
<tr>
<th>Technology Trigger</th>
<th>Peak of Inflated Expectations</th>
<th>Trough of Disillusionment</th>
<th>Slope of Enlightenment</th>
<th>Plateau of Productivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Open-Source Application Life Cycle Management Tools</td>
<td>Open-Source Content Management</td>
<td>Open-Source Message-Oriented Middleware</td>
<td>PC Linux for Consumers (Emerging Markets)</td>
<td></td>
</tr>
<tr>
<td>Open-Source Enterprise Service Bus</td>
<td>PC Linux for Consumers (Mature Markets)</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open-Source Business Intelligence Tools</td>
<td>Linux on 64 Processors</td>
<td>Open-Source AD Design Tools</td>
<td>Open-Source AD Testing Tools</td>
<td></td>
</tr>
<tr>
<td>Open-Source Virtualization</td>
<td>Open-Source DBMS</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open-Source Technology for Extreme Transaction Processing</td>
<td>Open-Source Service Management Tools</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open-Source Service Management Tools</td>
<td>Open-Source Social Software</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open Source for Customer Relationship Management</td>
<td>Open-Source Social Software</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open-Source E-Commerce Software</td>
<td>Open-Source Social Software</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open-Source Data Integration Tools</td>
<td>Open-Source Social Software</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open-Source Application Life Cycle Management Tools</td>
<td>Open-Source Social Software</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open-Source Application Life Cycle Management Tools</td>
<td>Open-Source Social Software</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
<tr>
<td>Open-Source Application Life Cycle Management Tools</td>
<td>Open-Source Social Software</td>
<td>Linux on 16 to 64 Processors</td>
<td>Linux on Desktop for Mainstream Business Users</td>
<td></td>
</tr>
</tbody>
</table>

As of July 2008

Source: Gartner (July 2008)

| Years to mainstream adoption: | less than 2 years | 2 to 5 years | 5 to 10 years | more than 10 years | obsolete before plateau |

Source: Gartner (July 2008)
### Table 1. Hype Cycle Phases

<table>
<thead>
<tr>
<th>Phase</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Technology Trigger</strong></td>
<td>A breakthrough, public demonstration, product launch or other event generates significant press and industry interest.</td>
</tr>
<tr>
<td><strong>Peak of Inflated Expectations</strong></td>
<td>During this phase of overenthusiasm and unrealistic projections, a flurry of well-publicized activity by technology leaders results in some successes, but more failures, as the technology is pushed to its limits. The only enterprises making money are conference organizers and magazine publishers.</td>
</tr>
<tr>
<td><strong>Trough of Disillusionment</strong></td>
<td>Because the technology does not live up to its overinflated expectations, it rapidly becomes unfashionable. Media interest wanes, except for a few cautionary tales.</td>
</tr>
<tr>
<td><strong>Slope of Enlightenment</strong></td>
<td>Focused experimentation and solid hard work by an increasingly diverse range of organizations lead to a true understanding of the technology's applicability, risks and benefits. Commercial off-the-shelf methodologies and tools ease the development process.</td>
</tr>
<tr>
<td><strong>Plateau of Productivity</strong></td>
<td>The real-world benefits of the technology are demonstrated and accepted. Tools and methodologies are increasingly stable as they enter their second and third generations. Growing numbers of organizations feel comfortable with the reduced level of risk; the rapid growth phase of adoption begins. Approximately 20% of the technology’s target audience has adopted or is adopting the technology as it enters this phase.</td>
</tr>
</tbody>
</table>

**Years to Mainstream Adoption**
The time required for the technology to reach the Plateau of Productivity.

*Source: Gartner (July 2009)*

### Table 2. Benefit Ratings

<table>
<thead>
<tr>
<th>Benefit Rating</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transformational</td>
<td>Enables new ways of doing business across industries that will result in major shifts in industry dynamics</td>
</tr>
<tr>
<td>High</td>
<td>Enables new ways of performing horizontal or vertical processes that will result in significantly increased revenue or cost savings for an enterprise</td>
</tr>
<tr>
<td>Moderate</td>
<td>Provides incremental improvements to established processes that will result in increased revenue or cost savings for an enterprise</td>
</tr>
</tbody>
</table>
### Benefit Rating Definition

**Low**

Slightly improves processes (for example, improved user experience) that will be difficult to translate into increased revenue or cost savings

**Source:** Gartner (July 2009)

---

### Table 3. Maturity Levels

<table>
<thead>
<tr>
<th>Maturity Level</th>
<th>Status</th>
<th>Products/Vendors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embryonic</td>
<td>• In labs</td>
<td>• None</td>
</tr>
<tr>
<td>Emerging</td>
<td>• Commercialization by vendors</td>
<td>• First generation</td>
</tr>
<tr>
<td></td>
<td>• Pilots and deployments by industry leaders</td>
<td>• High price</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Much customization</td>
</tr>
<tr>
<td>Adolescent</td>
<td>• Maturing technology capabilities and process understanding</td>
<td>• Second generation</td>
</tr>
<tr>
<td></td>
<td>• Uptake beyond early adopters</td>
<td>• Less customization</td>
</tr>
<tr>
<td>Early mainstream</td>
<td>• Proven technology</td>
<td>• Third generation</td>
</tr>
<tr>
<td></td>
<td>• Vendors, technology and adoption rapidly evolving</td>
<td>• More out of box</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Methodologies</td>
</tr>
<tr>
<td>Mature mainstream</td>
<td>• Robust technology</td>
<td>• Several dominant vendors</td>
</tr>
<tr>
<td>Legacy</td>
<td>• Not appropriate for new developments</td>
<td>• Maintenance revenue focus</td>
</tr>
<tr>
<td></td>
<td>• Cost of migration constrains replacement</td>
<td></td>
</tr>
<tr>
<td>Obsolete</td>
<td>• Rarely used</td>
<td>• Used/resale market only</td>
</tr>
</tbody>
</table>

**Source:** Gartner (July 2009)

---

### RECOMMENDED READING

"Understanding Gartner’s Hype Cycles, 2009"

"The Role and Impact of OSS in a Down Economy"

"Open-Source Vendor Strategies: Where OSS and Commerce Meet"

"Common Questions About OSS From IT Managers"

"The Role and Impact of OSS in a Down Economy"

This research is part of a set of related research pieces. See "Gartner's Hype Cycle Special Report for 2009" for an overview.