



**Open Source
vs Proprietary
Integration Platform:**
A decision-making
guide for enterprises

**Technical
Article**



Introduction

Enterprise Integration Platforms are moving faster to the top of IT wish lists. They offer the ability to use Application Programming Interfaces to integrate on-premises, cloud and mobile applications. Neither open source nor Proprietary software is a revolution in the industry. They have been there for some time. However, all technology decision makers face a critical decision when it comes to choosing between an open-source and licensed integration platform.

Implementing Enterprise Application Integration (EAI) can be an extremely expensive undertaking and is also a long term strategic move. They might need to purchase several products within the Integration Platform stack like an Enterprise Service Bus (ESB), API Manager, a portal and a data services tool. Well, it doesn't stop there. There are additional tools for testing, security and the list goes on and on. Therefore, it is advisable to pre-define needs, and then evaluate the products best suited.

The idea here is not to frame a war between open source and licensed software but seriously contemplate which one to be chosen and used appropriately. Proprietary solutions are often very similar, and also the most used, equally Open source competitors offer similar characteristics. It definitely makes sense to think of at the beginning, whether a proprietary or an open source solution is a better choice.



Here are some effective criteria points to be pondered to arrive at a decision.

Usability

How complicated is the installation? How many tools are needed? Is the development environment intuitive?

Maintainability

How do you administer the product? Is there a GUI for monitoring services?

Community

Are there active public forums or mailing lists? Are numerous articles, tutorials, articles, and videos available? Is the product supported by several companies?

Enterprise Support

What support options are offered ("business hours", "24/7" hotline vs. Email vs. on-site support, etc.)? Can the required service level agreements be guaranteed? Is support offered in your preferred language?

Functionality

Are all the required functionalities offered?

Flexibility

Can we customize functionalities of the product to fit our needs?

Expandability

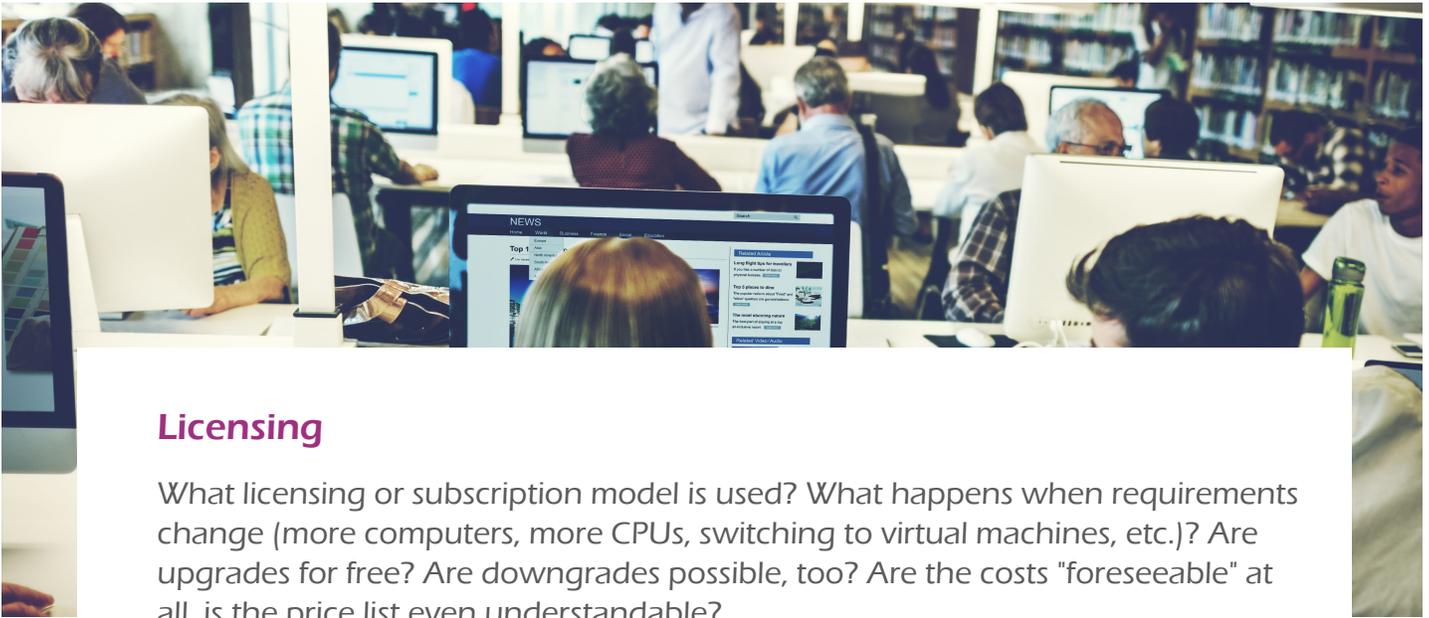
Is it possible to expand the product? Are the product and its interfaces based on standards?

Connectors

Are adapters for all required technologies available? Are there adapters for B2B products such as SAP or Salesforce? How easily can we build your own adapter?

Cost

What is the full cost (total cost of ownership) of the product - including maintenance, all required ancillary products, connectors, etc.?



Licensing

What licensing or subscription model is used? What happens when requirements change (more computers, more CPUs, switching to virtual machines, etc.)? Are upgrades for free? Are downgrades possible, too? Are the costs "foreseeable" at all, is the price list even understandable?

Why Open Source?

While what has been mentioned above help the decision makers to consider as factors for evaluation, let us now see the advantages of leveraging open source to meet EAI needs.

1 Try before you Buy

With commercial software, once we invest in the software, the cost of the software and the first year of maintenance are committed. With open source, we can create prototypes and try out the software before we commit large sums of money. Even if we plan on buying commercial software we can leverage open source software to validate our architecture before making any purchases.

2 Lower Cost of Entry

The cost of the various tools within the stack can be quite staggering. Open source eliminates or greatly reduces the initial sticker shock. We get the software for free and we have the option to subscribe for support services. Of course, not paying for support for our integration stack is not a wise choice. In some cases, there is a community version and an enterprise version. For certain products within the stack, we may only need the community version which is totally free. Other requirements may need the robust feature set of the enterprise version, but it is still substantially cheaper than commercial software.



3 Cost Effective Support

With commercial software, support costs a percentage of the initial purchase. This leads to incredibly high maintenance costs in the hundreds of thousands or even millions of dollars. Support for open source software is substantially less. So not only do we not have the huge initial investment, but our ongoing fixed costs are substantially less as well.

4 Core Competency

Many of the mega vendors in the integration platform space have tentacles in various areas of technology with integration platform being one of them. They often take existing products and tweak them in the name of EAI/Integration Platform. They also buy several companies and then call themselves an integrated stack. The reality is, their stacks are a jumble of many different companies and the promise of integration is not a reality. With many of the open-source vendors, ESB is all that they do. These products are built for ESB from the ground up, not from mergers, acquisitions, and rushed integration releases.

5 For the People

Commercial software is closed and not accessible by developers. In case of some integration challenges with a vendor's portal we need to provide the vendor with logs and various information so that they could troubleshoot the issues. Once they finally find the bugs, they might fix some of them but might defer others to a future release to be determined. If we have access to the code, we would fix issues sooner and save the precious time that we lost while waiting for the vendor to resolve its issues.

That might make you feel Open source to be inherently cool nevertheless it also comes with warning labels on commercial aspects. The ethos of open source is mainly driven by the self-governance principles and open web set for changes. Philosophically something worth to be applauded but makes the system highly vulnerable to unprofessional developers.



Why Licensed Version?

Keeping the fear factor aside that might get us spaced out, let us equally mull factors over licensed integration platform as well.

1 Open is not Free

While open-source solutions don't charge license fees, agencies may run up expenses because the solutions can be difficult to configure and customize. Often developers must manipulate large XML files and write complex code to support relatively straightforward business requirements. So, "free" may actually turn expensive if we don't have internal staff with a strong grasp of open-source environments, and specialists are required for installation and ongoing administrative tasks and support.

2 Long-term Growth

Investing in any new integration platform can be costly and time consuming. Therefore, we should be certain that what we choose will support scalability and long-term expansion plans. It is widely known that open-source ESBs simply cannot match the scale and number of integration implementations of a commercial ESB solution. As such, they are usually deployed at small companies or in small, developer-based projects at larger enterprises. In addition, we must consider vendor viability. In a constantly changing industry, open-source vendors may be acquired, merged or fail, leaving customers with no continuity.

3 Integration Tolerance

Open-source software typically requires extensive integration during implementation. This means an open-source ESB often provides limited options for integrating into systems managing business processes, file transfer and APIs. Also, it is important to consider the depth of adapter offerings provided by the ESB vendor, especially for packaged applications. Not having the adapters that we need impacts developers' productivity because they may have to write custom code.



4 Support Options

Whether we are leaning towards licensed or open-source integration options, a support contract is critical. This support is necessary to avoid deteriorating software quality, security vulnerabilities, patches that may not get installed, bugs that may be left unfixed and, ultimately, an increase in overall total cost of ownership. In an open-source environment, we may be forced to find the necessary support skills internally or via a support contract with a specific open-source ESB provider.

5 External Partners

Integration is not always limited to internal applications and IT systems. It might also include external entities, such as customers, suppliers and vendors. We must be prepared to have B2B solutions in place that will work closely with the selected integration solution.

6 Messaging Requirements

Messaging often gets overlooked in a Licensed vs. open-source ESB decision. Evaluate the messaging needs and make sure the chosen provider offers best-of-breed, “universal” messaging support across the various delivery channels and implementation topologies currently in use. Be sure open-source messaging software can support the high-speed and high-volume requirements posed by modern-day applications.

7 Control of the Product Roadmap

Open-source integration software is built by a community of developers. As a result, individual open-source vendors can have little or no control over the developer roadmap. The future of the product may or may not be in line with our user requirements. This could become a great issue, if we have key initiatives that must be executed by a specific time.

8 iPaaS Limitation

With the growing popularity of Integration Platform as a Service (iPaaS) more and more enterprises are moving towards cloud-based iPaaS model. An enterprise can achieve a significant return on investment by choosing this solution rather than using a corporate-owned data center. Small open-source ESB providers may not be able to support all integration categories, forcing them to invest more heavily in supporting a cloud-based model. As a result, they may pass those investment costs directly on to customers, further closing the gap in the cost differential between open-source integration platform and CoTS solutions.

9 API Management

Full life cycle API management is about the planning, design, implementation, publication, operation, consumption, maintenance and retirement of APIs. It includes a developer's portal to target, assist and govern the communities of developers who embed the APIs, as well as the runtime management and analytics. Full life cycle API management is the functionality organizations need in order to provide the technology platform for digital business and run successful API programs. There are very few players in the open-source market that provide API Management solution with all round capabilities. Finding one complete open source API tool that caters to all our needs is a difficult task at present.



Conclusion

Proprietary solutions generally offer more features and "powerful" support. However, the question remains, "Are these really needed?" Remember that effort, complexity, and costs are correspondingly higher. Open Source products score with easier usability, greater flexibility, easier extensibility and lower costs. There are a few paths that we can take for our open-source ESB initiative. We can go with a complete open-source integration stack, we can mix and match various open-source integration products from different vendors, or we can mix and match both commercial and open-source products.



About Aspire Systems

Aspire Systems is a global technology services firm serving as a trusted technology partner for our customers. We work with some of the world's most innovative enterprises and independent software vendors, helping them leverage technology and outsourcing in our specific areas of expertise. Our core philosophy of "Attention. Always." communicates our belief in lavishing care and attention on our customer and employees.

SINGAPORE
+65 3163 3050

NORTH AMERICA
+1 630 368 0970

EUROPE
+44 203 170 6115

INDIA
+91 44 6740 4000

MIDDLE EAST
+971 50 658 8831

For more info contact
info@aspresys.com or visit www.aspiresys.com

