

Guidelines for Assessment of Clinical Trial Software Integration Capabilities

CONTENTS

Introduction	2
“Integration” vs “Interface”	2
Integration Scope.....	3
Base Integration Scenario	4
Integration Technologies	5
Graphical Workflow – Further Defined	6
Graphical Integration Options	7
Critical IT Support of Clinical Trial Software Application.....	9
Considerations	10
Conclusion	11

Introduction

It's hard to find a vendor who is not touting the integration capabilities of their clinical trial applications. Evaluators of clinical trial offerings generally recognize the importance, and usually at least check to see that the "integration" word appears somewhere in the vendor's sales material. However, integration capabilities differ significantly among clinical trial software vendors, with serious ramifications for the IT department faced with supporting the system as part of an enterprise over a product life-cycle that may span decades. This paper tries to provide an IT department with factors that should be considered in the evaluation of any clinical trial software system, and translates these into business impact terms for communication with the business owners to help make the best overall decision for the company.

"Integration" versus "Interface"

The first step in evaluating vendor offerings in terms of integration capabilities is whether or not they actually support integration, or if their applications' capabilities are limited to interfacing. For the purposes of this paper, the following distinction will be made:

Interface Capability – Makes internal data available in some documented form for use by external applications.

Integration Capability – The vendor provides software applications, tools or functionality that supports the development of integration software to facilitate the interconnection of disparate software.

So, the important question is not simply whether data is available from the application, but what is going to be needed to support the integration of this application in a dynamic enterprise over a lifespan that can potentially span decades. The answer to this question has tremendous bearing on costs, responsiveness to changing business needs and the internal skill types that will be necessary to provide the level of support demanded of the modern IT organization.

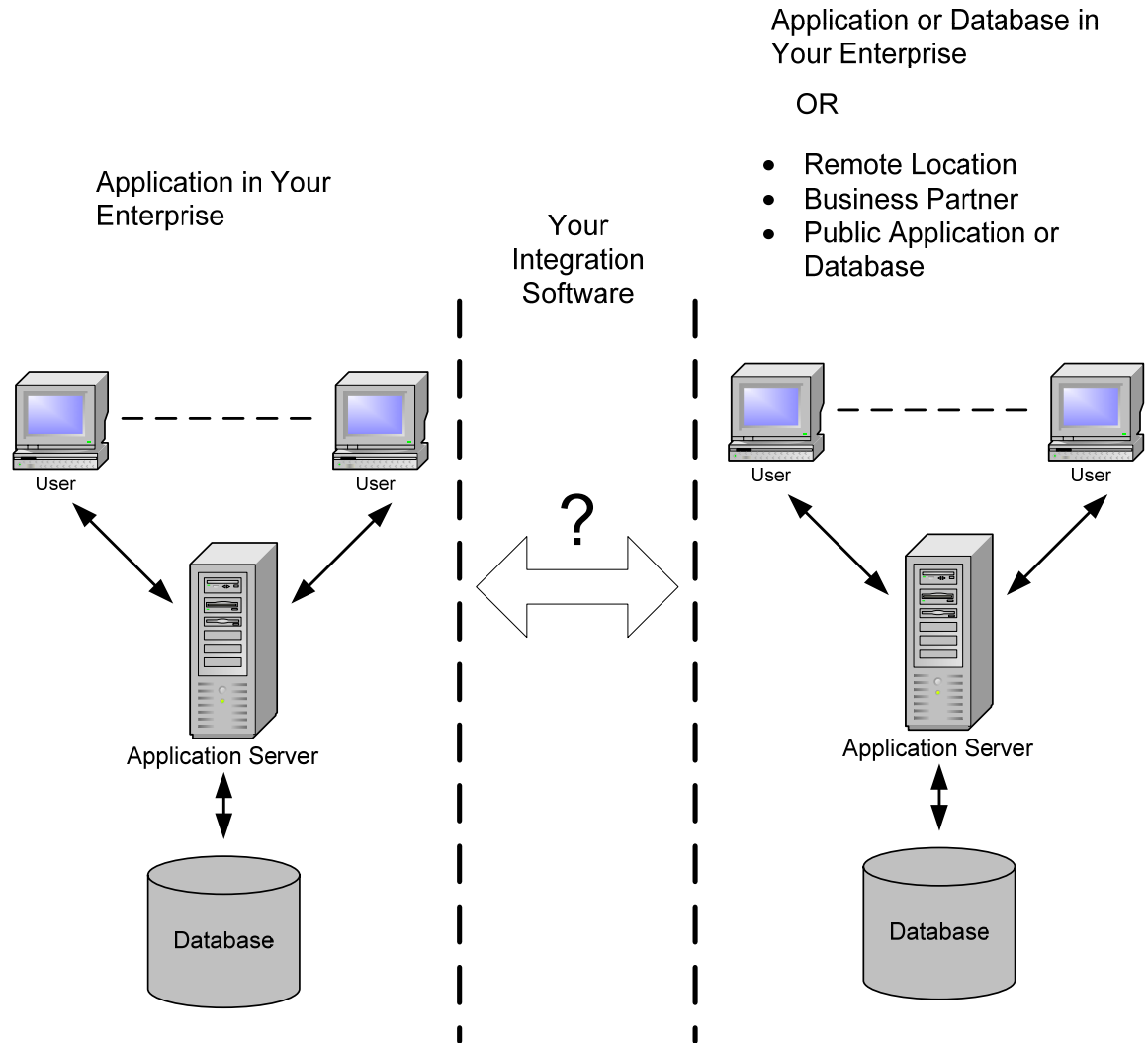
The remainder of this paper focuses upon the most common methods of achieving integration between applications, and provides the primary considerations in terms of business impacts.

Integration Scope

A key consideration is the scope of integration addressed by the vendor’s applications. It’s important to take a deep look at how various current and future integration needs might be met. The following table serves as a comparison checklist, in order of increasing integration scope:

Scope of Integration	Vendor A	Vendor B
Integration within all vendor offerings currently under consideration, as well as those that could potentially be installed in the enterprise.		
Ability to integrate with existing or potential applications from other clinical trial software vendors. Can adapt easily to evolving standards like CDISC.		
Architecture that facilitates integration with remote business systems supported by the company IT department.		
Addresses integration with existing non-clinical systems, or software that could be installed in the enterprise in the future (Examples - ERP, other financial and manufacturing systems).		
Web based architecture that facilitates integration with business partner clinical trial systems, or other software systems.		

Base Integration Scenario



The Question Is:
“How does one get these systems to work with each other?”

Integration Technologies

Vendors offer many technologies to try to meet integration needs. Unfortunately, some of these offerings may also be tied to marketing strategies that might limit a company's ability to switch vendors, or to purchase from other vendors in the future. The various integration technology options need to be fully examined in view of the key business drivers of cost and schedule.

Here's a brief overview of the integration technology landscape:

Integration Technology	Description
Application Program Interface (API) Based	Interfaces to internal software functions or "methods", are documented and made available for access within various software development platforms. Companies can implement integration capabilities by developing software that invokes vendor software via the defined API. An API may provide capabilities beyond integration, allowing the user to utilize other functionality within the application.
Connectors/Adaptors	Similar to an API, but the vendor provides specific integration modules that can be incorporated within user developed software.
Web Services Based	Basically, an API that is based upon an industry standard for exchanging data over the web. Although the communication mechanism is standard, some programming is generally necessary to generate and accept data. However, modern development environments simplify the software development process.
Database Standards	Standards that allow access to different database management systems. The most common standard is ODBC, which stands for Open Database Connectivity. Although the method for interchange of data is standardized, programming is necessary to control the data interchange.
Graphical Workflow	A more recent software technology where common software functions, or process modules, can be "dragged and dropped" and connected in the desired operational sequence via a graphical editor.
Graphical Workflow with Software Development Kit (SDK)	An SDK allows the user to expand the capabilities of the graphical workflow platform through development of new process modules to accommodate business functions or integration modules now provided by the software vendor.

Graphical Workflow - Further Defined

A graphical workflow approach supports the development of integration software without code development, and also provides powerful capabilities that can orchestrate the flow of data between applications and users based upon events or time.

Integration Technology	Description
Graphical Workflow	A more recent software technology where common software functions, or process modules, can be "dragged and dropped" and connected in the desired operational sequence via a graphical editor.

While graphical workflow provides the ability for a user with little software expertise to generate the integration workflow, some IT organizations will want to empower higher skilled programmers with the ability to actually develop the process modules, now or in the future. In this case, the IT organization should look for a vendor that offers a Software Development Kit, or SDK, that provides libraries, templates and tools for the development of process modules.

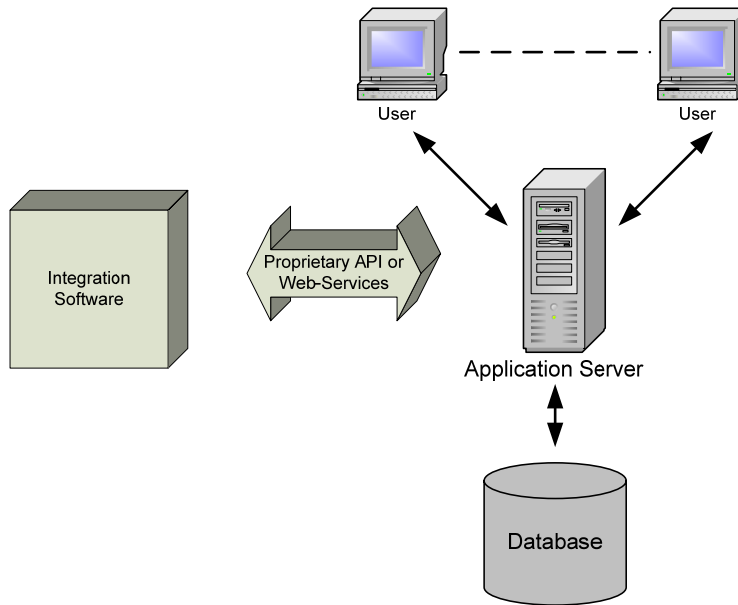
As an example, the IT organization may be tasked with developing software that integrates with another internal application, like an MS Access Database or an internal financial system. A more senior developer can develop the process module using the software language of choice, and then make the process module available as part of a library within the graphical workflow environment.

Integration Technology	Description
Graphical Workflow with Software Development Kit (SDK)	An SDK allows the user to expand the capabilities of the graphical workflow platform through development of new process modules to accommodate business functions or integration modules now provided by the software vendor.

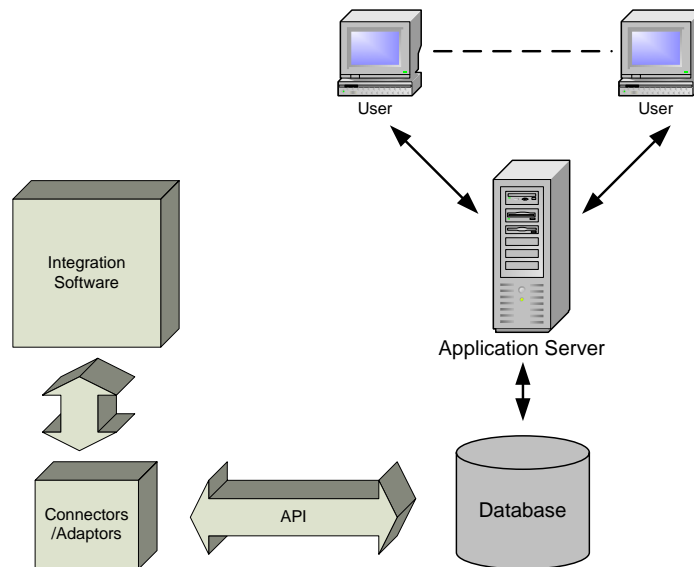
Graphical Integration Options

The following diagrams provide a simplified view of the integration options that interconnect integration software and applications:

Proprietary API or Web-Services Integration:

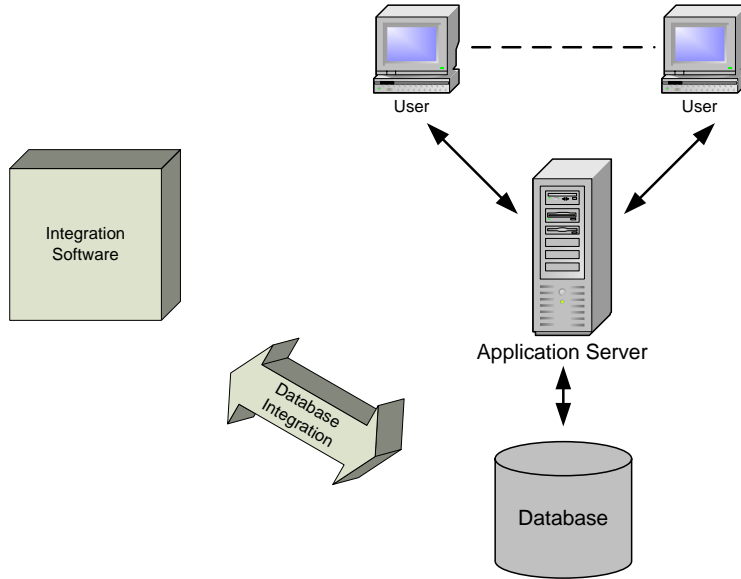


Connectors/Adaptors:

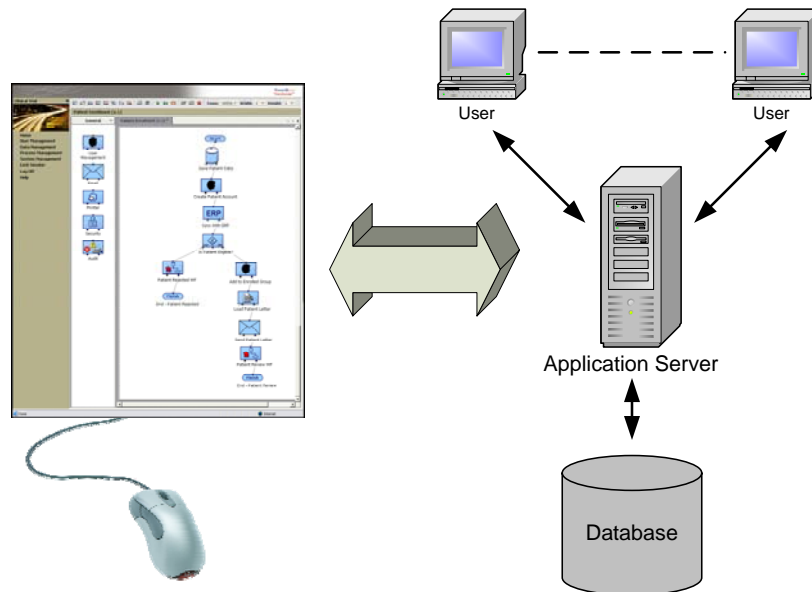


Graphical Integration Options – continued

Database Integration:



Graphical Workflow Integration:



Critical IT Support of Clinical Trial Software Application

What does it all mean to the IT organization tasked with supporting a clinical trial software application?

While the purchase of a clinical trial application may be driven from the clinical operations or data management side, it is the IT department that is faced with supporting requests for integration over the life-cycle of the installation. It's not reasonable to expect that those primarily responsible for the selection and purchase of such software to fully understand the implications and challenges of integration within an evolving enterprise system. Therefore, the IT department needs to carefully evaluate the life-cycle integration implications of vendor offerings, and effectively communicate them to the business process owners. The following categories are key considerations that should be considered when assessing vendor integration capabilities:

Flexibility

How easily can the installation respond to changing business needs and regulatory requirements? Is there a good chance that the system can adapt to yet unforeseen integration requirements? Flexibility is especially important for small to mid-size life sciences companies where there is anticipation for rapid growth and rapidly changing requirements through drug commercialization phases, and the complete needs of the enterprise are difficult to predict.

Responsiveness of the IT Organization

How quickly can the IT department respond to new integration requests from the business owners? Does it involve a classical development cycle, or can changes be made "on the fly"? IT departments are often given little or no advance warning of new business requirements, and are typically left "holding the ball" on any delays on the roll-out of critical business changes. Can changes be implemented rapidly without sacrificing the ability to expand integration in the future?

Organizational Technical Capability

What is the technical depth required of the IT department to support changes to integration needs? Are object-oriented software development skills necessary, or can a person with very basic programming skills make changes? This impacts IT labor and training costs, as well as the ability to obtain resources.

Internal Development Cost

What is the development cost to support a typical integration request? This impacts IT budgets over a product life-cycle that might span decades.




Vendor Independence

To what extent will the company be dependent on the software vendor to make integration changes? Increased dependence can lead to reduced pricing leverage and poor support. This is also related to risk. What happens if the vendor decides not to support integration for your particular needs? Will you be stranded? Also, to what extent do the integration capabilities enhance the ability of the IT organization with multi-vendor options for selection of future applications?

TranSenda White Paper

The following table provides a view of the various integration technologies relative to these assessment categories:

	Graphical Workflow w/SDK	Graphical Workflow	Database Integration	Web Services	Connectors/ Adaptors	API
Flexibility	Excellent	Satisfactory	Satisfactory	Satisfactory	Satisfactory	Excellent
Responsiveness of IT Organization	Excellent	Excellent	Satisfactory	Satisfactory	Higher Risk	Satisfactory
Organizational Technical Capability	Excellent	Excellent	Satisfactory	Satisfactory	Satisfactory	Higher Risk
Internal Development Cost	Excellent	Excellent	Satisfactory	Satisfactory	Satisfactory	Higher Risk
Vendor Independence	Excellent	Satisfactory	Satisfactory	Satisfactory	Higher Risk	Higher Risk

Excellent	
Satisfactory	
Higher Risk	

Considerations

- While an API provides a high degree of flexibility, this type of integration model needs the highest skill levels, usually requiring internal developers with experience in object-oriented design techniques and software development languages like C++ or Java. Internal development and support costs are higher because classical software development life-cycles spanning requirements, detailed design, coding and testing phases are required. Vendor introduced changes to the API can ripple through legacy integration software. Vendor dependence is high because the IT department is totally dependent upon what the vendor decides to expose and maintain in terms of the API.
- Connectors and adaptors have somewhat less flexibility than a general API in that a finite set of integration options is available. Vendor dependence is still high because availability of specific connectors or adaptors is controlled by the vendor. Simpler programming languages, like Visual Basic, can be used, but a standard software life-cycle development cycle is still required.
- Database or web services integration is generally superior to an API or connector/adaptor approach in that there is some level of standardization. However, standards like ODBC and web services provide standards at the message transport level, and are not totally independent of the database contents or application methods.
- A graphical workflow approach provides a much simpler platform for integration changes, with the added benefit that business rules can be added or customized “on the fly” by people with minimal programming skill. However, the available set of integration modules is still generally controlled and sold separately by the vendor.
- Adding an SDK to a graphical workflow platform provides the best of all options. The SDK empowers the IT department with the ability to create their own integration modules, which can then be linked into workflow by people with lower programming skills. Standard process modules can be purchased from the vendor, but the IT department retains the ability to develop new modules internally. These modules can be developed with high power programming languages like C++ or C#, or simpler languages like Visual Basic, depending upon the level of sophistication of the IT department’s development capability.

Conclusion

Selection of any modern-day clinical trial applications should involve a serious evaluation of integration capabilities. This requires a deeper look at the various options that software vendors provide for integration, and an understanding of the business implications of each of these integration models.

A successful evaluation requires collaboration among the clinical trial process owners and the IT department responsible for the integration and evolution within today's dynamic enterprise.