



**The Return on Investment
of Open Source Data Integration**

WHITE PAPER

Table of Contents

Table of Contents	2
Open source, a model that benefits all parties	3
The different alternatives open source data integration is replacing.....	5
Elements of the ROI.....	6
ROI calculations	8
Cost Comparison and ROI	17
Summary & Conclusion	19

Open source, a model that benefits all parties

Cost is often presented as one of the major reasons for adopting open source solutions for IT projects. However, cost savings must be determined carefully, as the actual cost of a project goes well beyond the obvious initial costs - the license costs, which can be viewed as the tip of the iceberg - and include a much greater component, the hidden part of the iceberg.

It is not the purpose of this white paper to provide a tutorial on the cost calculation of an IT project. Rather, this white paper will explain the different alternatives to open source data integration, and will present the costs - visible and hidden - of these alternatives, on several types of projects.

Before getting into the core subject of this white paper, it is worth mentioning a few facts about the business model of open source vendors – and killing several false ideas at the same time.

- The *free* part of open source is not related to the absence of costs but to the freedom to use the solution (to use a common analogy, it's "free" as in "free speech", not as in "free beer").
- Open source vendors are for-profit organizations, often backed by reputable financial firms, who expect a return on their investment.
- Although there are variants to this model, open source vendors often provide a no-cost solution to the community and charge for services (support, training, expertise, etc.) and value-added features, needed only by enterprises for large-scale projects.
- However, even the services and value-added features are significantly less expensive than what vendors of comparable proprietary solutions would charge for them. There are many reasons for this lower cost, among which one can mention:

- A sharply decreased cost of sales, due to the broad community adoption (oftentimes, projects start small with the free version, and grow).
 - The involvement of the community in the design, development and testing of the solution decreases the cost of development.
 - Open source vendors, being usually newer companies, don't drag along years and years of legacy compatibility that they have to maintain.
- Open source vendors are not "vampires" who take advantage of the free work of the community. Vendors do give back a lot of features to community users, at no charge. And by the way, one should never assume open source community members are naive and are taken advantage of – they are not! The open source model is actually beneficial to all parts involved: community members, paying customers, and vendors.

The different alternatives open source data integration is replacing

Open source is not the only approach to data integration. Other approaches used by many organizations include on-demand custom development, and proprietary data integration solutions.

On-demand custom development, a.k.a. "manual coding"

"We only had three months to develop our integration solution - actually, more like 5 to 6 weeks when accounting for tests and deployment. Custom scripts development could not possibly have worked, both in terms of timing and in terms of features."
-- Medialab

On-demand custom development, also called manual coding, is perhaps the most widely used approach to data integration. Simply put, it consists of writing programs in third generation programming languages (C++, Java, etc.), or in scripting languages (Perl, Python, etc.), to address data integration needs. These programs would typically extract data from one or several sources, perform any number of data processing and transformations, and then write the data to its target(s). While offering a great degree of flexibility, the manual coding approach is time-consuming, requires high levels of programming skills, and poses great technical difficulties for any process that goes beyond the simplest ones. In addition, manual coding creates maintenance headaches, familiar to anyone who has had to maintain code written by other developers.

Proprietary data integration solutions

"We have quickly been pushed to abandon [Oracle's] solution (...): our growth model turned out to be incompatible with the new pricing model of Oracle, based on a cost per target CPU."
-- Eurofins

Unlike manual coding, proprietary data integration solutions provide an abstraction layer that decreases the complexity of the developments, and offer much higher productivity. However, proprietary data integration solutions are plagued with high costs (both visible and hidden), long learning curves, and do not offer a high degree of flexibility.

Elements of the ROI

The cost studies presented here take into account a number of elements. Some of them can easily be quantified, while others are difficult to evaluate and are mentioned here as intangibles.

- **Development licenses** costs include the acquisition costs of development licenses. They apply only to proprietary data integration solutions.
- **Training curve** costs include both quantifiable elements – the costs of training classes and the salaries of the staff during their training and rampup phases – and intangible elements: the time to market of the project, which is delayed by lengthy training. They apply both to proprietary and open source data integration solutions (for the sake of argument, this white paper considers that developers performing manual coding are already proficient in the language they use).
- **Development time** includes both quantifiable elements – the salaries of the staff during the development phase of the project – and intangible elements: the time to market of the project. They apply to all approaches.
- **Run-time licenses** costs include the acquisition costs of the run-time licenses (engine, connectors, etc.). They must be paid when the project goes into production, and every time an extension to the project is performed: introduction of new sources/targets, increase in the processing power (number of CPUs...) due to data volume increases, etc. They apply only to proprietary data integration solutions.
- **Deployment hardware and operating system licenses** (and applicable maintenance) apply to the systems used to run the data integration processes. They apply to all approaches.

"The open source model limited the need for an upfront investment, an important factor in a fiscally responsible organization, financed by taxpayers' money."
-- City of Brantford

- **IT operations** costs include the salaries of the staff that monitors the proper execution of the data integration jobs once they have been deployed in production. They apply to all approaches.
- **Maintenance/subscription** costs include the ongoing costs paid to the vendor for benefiting from support, maintenance, and subscription features. They apply both to proprietary and open source data integration solutions.
- **Maintenance time** includes both quantifiable elements – the salaries of the staff during the maintenance phases of the project – and intangible elements: the time to market of new features and corrections. They apply to all approaches.
- The **reliability and predictability** of the data integration processes are intangible but critical elements for the success of projects. Data integration processes that do not consistently run without error, or fail to deliver accurate and timely data, will do little for the success of business operations. These elements apply to all approaches.

ROI calculations

This section of the white paper presents several return-on-investment calculations, for three different projects: one "small", one "medium" and one "large". "Small", "medium" and "large" apply to the number of systems involved, the number of data integration processes (data flows) required, and the complexity of the project.

For each project, this white paper estimates the costs associated with each of the three approaches: manual coding, proprietary data integration, and open source data integration. It also comments on the intangible elements of each approach.

Small project: data migration

The "Small" project is a relatively simple data migration project. It consists of migrating customer data contained in a RDBMS, to a CRM application. Data transformations and lookups in files (for enrichment) need to be performed as part of this migration. It is a one-time migration (no on-going maintenance). This project involves 20 data flows.

Approach: manual coding		
Development time	1 developer, 8 weeks	\$12,000
Run-time server	Not required (run on existing systems)	\$0

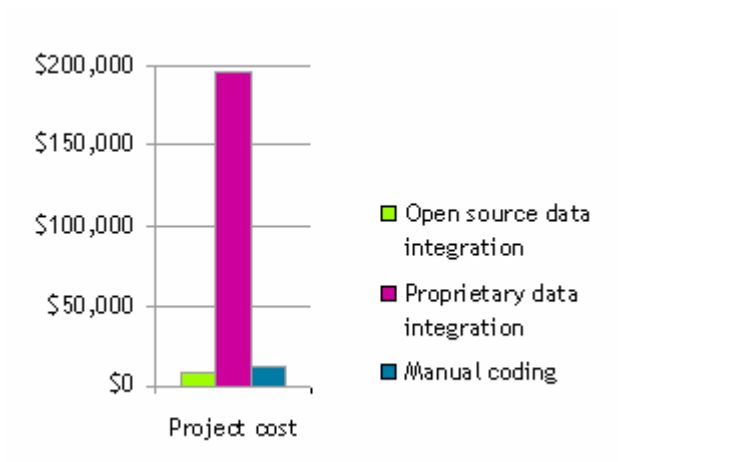
Approach: proprietary data integration		
Development license cost	1 developer seat with 1 year maintenance	\$36,000
Training	1 developer, 1 week (cost of class + salary)	\$5,500
Development time	1 developer, 2 weeks (salary)	\$3,000
Run-time license cost	1 run-time engine 1 RDBMS connector 1 CRM connector 1 year maintenance on the above	\$150,000
Run-time server	Intel server with 1 dual-core CPU running Windows 2003 Server	\$1,500

"We looked at the various alternatives on the market. Talend Open Studio was offering several advantages. First of all, the free license was attractive in a several ways: budgetary savings, and no license management."
-- European Commission Joint Research Center

Approach: open source data integration (Talend Open Studio)		
Subscription cost	1 developer seat (includes Gold support)	\$2,150
Training	1 developer, 3 days (cost of class + salary)	\$2,700
Development time	1 developer, 2 weeks (salary)	\$3,000
Run-time server	Not required (run on existing systems)	\$0

Summary

Approach	Project cost	Time-to-market
Manual coding	\$12,000	8 weeks
Proprietary data integration	\$196,000	3 weeks
Open source data integration	\$7,850	3 weeks



Graph 1 – Total cost for a small project, comparing the use of 3 approaches to data integration: open source, proprietary and manual coding

Intangible elements

In addition to the costs elements highlighted above, time-to-market is an important factor for this type of project. The use of a tool accelerates greatly the speed of development, and hence the availability of the new system.

Medium project: ETL

The "Medium" project is an ETL project that consists of loading daily a data warehouse with data contained in three different RDBMS and a CRM application, and data provided by Web Services. Data denormalizations and lookups against reference files are performed. This project involves 120 data flows.

Approach: manual coding		
Development time	5 developers, 24 weeks	\$180,000
Run-time server	2 Intel servers, each with 2 dual-core CPUs, running Linux	\$4,000
Run-time server operation & maintenance	Yearly	\$1,000
Maintenance time	1 developer, 4 weeks per quarter (salary) - yearly	\$24,000

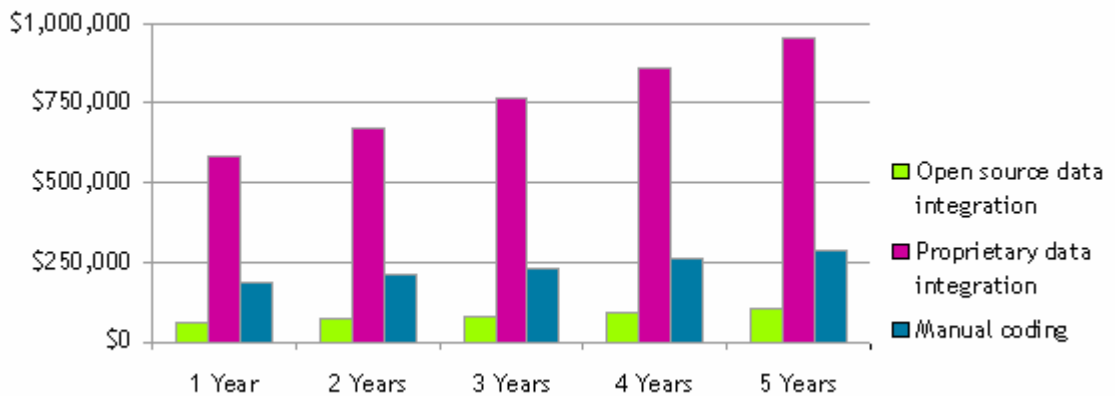
Approach: proprietary data integration		
Initial development license cost	3 developer seats with first year maintenance	\$90,000
Ongoing development license cost	Yearly maintenance	\$15,000
Training	3 developers, 2 weeks (cost of class + salary)	\$33,000
Development time	3 developers, 8 weeks (salary)	\$36,000
Run-time server	1 Intel server with 4 dual-core CPUs, running Windows Server 2003	\$10,000
Run-time server operation & maintenance	Yearly	\$2,500
Initial run-time license cost	1 run-time engine (8 cores) 4 RDBMS connectors 1 CRM connector 1 Web Services connector First year maintenance on the above	\$420,000
Ongoing run-time license cost	Yearly maintenance	\$70,000
Maintenance time	1 developer, 1 week per quarter (salary) - yearly	\$6,000

"We never could have created manually the thousands of lines of code that Talend Open Studio generated automatically. This allowed us to spend more time on data than on 'plumbing' and save us lots of valuable time."
-- Easyssur

Approach: open source data integration (Talend Integration Suite, Team Edition)		
Subscription cost	3 developer seats (includes Gold support) - yearly	\$8,500
	1 maintenance seat (includes Gold support) - yearly	\$4,000
Training	3 developers, 1 week (cost of class + salary)	\$13,500
Development time	3 developers, 8 weeks (salary)	\$36,000
Run-time server	2 Intel servers, each with 2 dual-core CPUs, running Linux	\$4,000
Run-time server operation & maintenance	Yearly	\$1,000
Maintenance time	1 developer, 1 week per quarter (salary) - yearly	\$6,000

Summary

Approach	Cost Year 1	Cost for 5 Years	Time-to-market
Manual coding	\$184,000	\$284,000	24 weeks
Proprietary data integration	\$581,500	\$955,500	10 weeks
Open source data integration	\$62,000	\$106,000	9 weeks



Graph 2 – Total project cost over 5 years for a medium project, comparing the use of 3 approaches to data integration: open source, proprietary and manual coding

Intangible elements

In addition to the costs elements highlighted above, the use of a tool greatly accelerates the time-to-market of the project. It also decreases the risk factor: tool-based developments are more accurate and reliable, and less error-prone than manual coding. Maintainability is also a significant benefit brought by a tool: not only does the use of a tool sharply decrease ongoing maintenance costs, it also speeds up the time-to-market of updates to the data integration processes.

In addition, should the scope of the project change, for example with additional data sources being added, or more processing power being required, open source brings predictability to the overall project costs, whereas proprietary software would require the acquisition of additional connector licenses, or run-time CPU licenses.

Large project: real-time enterprise data integration

The "Large" project is a complete enterprise data integration project. Twelve different databases, two packaged applications (one ERP and one SCM), an application from a partner (accessed through Web Services), a LDAP directory, and a Software-as-a-service CRM all need to be kept synchronized in real-time. This project involves 200 complex data flows.

Approach: manual coding		
Development time	40 developers, 24 months	\$5,760,000
Run-time server	6 Intel servers, each with 2 dual-core CPUs, running Linux	\$12,000
Run-time server operation & maintenance	Yearly	\$3,000
Maintenance time	10 developers full time (salary) - yearly	\$720,000
IT Operations time	2 developers full time (salary) - yearly	\$144,000

Approach: proprietary data integration		
Initial development license cost	15 developer seats with first year maintenance	\$300,000
Ongoing development license cost	Yearly maintenance	\$50,000
Training	15 developers, 2 weeks (cost of class + salary)	\$165,000
Development time	15 developers, 12 months (salary)	\$1,080,000
Run-time server	2 Unix servers, each with 4 dual-core CPUs	\$100,000
Run-time server operation & maintenance	Yearly	\$25,000
Initial run-time license cost	2 run-time engines (each for 8 cores) 12 RDBMS connectors 2 ERP/CRM connectors 1 Web Services connector 1 LDAP connector First year maintenance on the above	\$885,000
Ongoing run-time license cost	Yearly maintenance	\$147,500
Maintenance time	1 developer full time (yearly salary) - yearly	\$72,000
IT Operations time	1 developer full time (yearly salary) - yearly	\$72,000

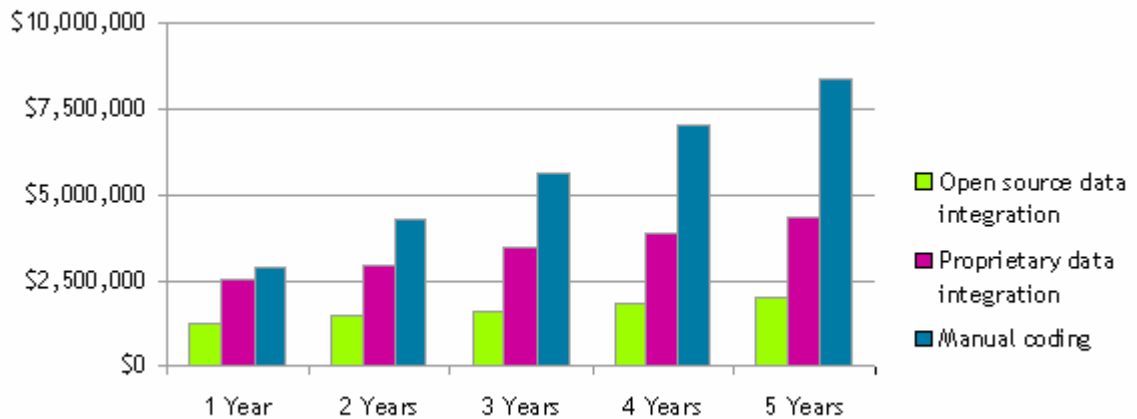
Approach: open source data integration (Talend Integration Suite, Enterprise Edition)		
Subscription cost	15 developer seats (includes Gold support) - yearly	\$108,000
	5 maintenance seats (includes Gold support) - yearly	\$38,000
Training	15 developers, 1 week (cost of class + salary)	\$67,500
Development time	15 developers, 12 months (salary)	\$1,080,000
Run-time server	4 Intel servers, each with 2 dual-core CPUs, running Linux (plus distributed processes on existing hardware)	\$8,000

"Talend was the only one to ensure us continuity and real transparency [of prices]. Moreover their prices were much more affordable than those from the other two software providers."
-- Naville

Run-time server operation & maintenance	Yearly	\$2,000
Maintenance time	1 developer full time (yearly salary) - yearly	\$72,000
IT Operations time	1 developer full time (yearly salary) - yearly	\$72,000

Summary

Approach	Cost Year 1	Cost for 5 Years	Time-to-market
Manual coding	\$2,892,000	\$8,373,000	24 months
Proprietary data integration	\$2,530,000	\$4,358,000	12 months
Open source data integration	\$1,263,500	\$1,999,500	12 months



Graph 3 – Total project cost over 5 years for a large project, comparing the use of 3 approaches to data integration: open source, proprietary and manual coding

Intangible elements

For this project, not only does the use of a tool greatly accelerate the time-to-market of the project, it also significantly decrease the ongoing maintenance costs. However on a project of this magnitude, the tool provides before all reliability and predictability. Manually-coded processes require constant attention and care from the IT Operations or-

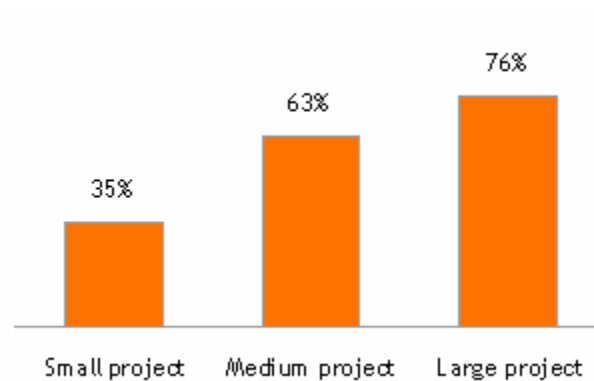
ganization, and they oftentimes fail to deliver the consistency that is needed by mission critical systems such as this one.

In addition, a project such as this one is almost never entirely scoped out at its start, and requirements for additional source and targets are often added as the project moves along. With proprietary data integration, that would mean additional, unpredictable costs. Open source adds cost predictability to the equation.

Cost Comparison and ROI

Open source vs. manual coding

Looking only at the actual cost savings, the return on investment provided by open source increases dramatically as the size of the project grows, as shown in graph 4.



Graph 4 – Cost savings induced choosing by open source over manual coding for data integration

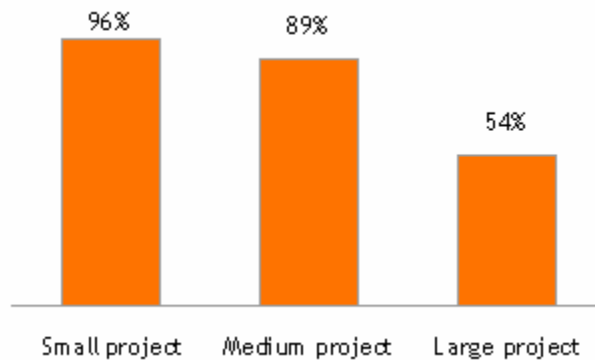
This ROI differences can be explained by several elements:

- Using a tool for small projects involves a learning curve (training), which is more easily amortized on larger projects
- Manual coding is a viable, albeit cumbersome approach for low complexity projects

However, as discussed before, a number of intangible factors also exist that tend to plague manual coding projects: delayed projects, unreliable processes, unpredictable execution, etc.

Open source vs. proprietary

Graph 5 below shows the return on investment brought by open source as compared to proprietary data integration technologies. For all three project typologies, open source brings a considerable ROI, which however varies vastly depending on the size and complexity of the project.



Graph 5 – Cost savings induced choosing by open source over proprietary data integration technologies

These variations are due to a number of factors:

- Proprietary data integration induce license costs and learning curve that are significantly higher than open source ones, making the upfront investment extremely steep for small and medium projects.
- As the complexity of the project grows, so does the amount of development, which starts to counterbalance the initial license and training investment.
- However, the open source licensing model still maintains a significant advantage over the total project life cycle, preserving a clear advantage for open source.

"Talend's Open Source model does not cancel all costs but it alleviates them significantly, especially in the deployment phase."
-- ETAI

Summary & Conclusion

The scenarios described in this paper illustrate the cost savings and the return on investment that open source bring to data integration projects:

- For small projects, manual coding is a viable solution but open source brings savings up to 35% of the total project costs and many intangible benefits. Proprietary technologies costs usually make them unpractical for such projects typologies.
- Medium projects really offer users the choice of the approach: open source, proprietary, or manual coding. Compared to proprietary technologies, open source savings can be up to 89%, whereas when compared to manual coding the savings reach 63%.
- In large projects, manual coding usually introduce too much complexity and lengthen time to market. In this scenario, open source costs savings reach 54% over the use of proprietary technologies.

In all cases, open source brings significant cost savings. However the benefits of open source reach well beyond these savings with many intangible factors that need to be considered as well.

For more information on Talend's open source data integration solutions: <http://www.talend.com>
Contact information for your region: <http://www.talend.com/contact>

© 2008 Talend Inc. All rights reserved.

Java is a registered trademark of Sun Microsystems, Linux is a registered trademark of Linus Torvalds. All other brand names or products referenced herein are acknowledged to be trademarks or registered trademarks of their respective owners.