ID		Participant 1	Participant 2	Participant 3	Participant 4	Participant 5	Participant 6	Participant 7
1	**Do you have any experience with integration using a ESB?**	Yes	Yes	Yes	Yes	Yes	Yes	Yes
1.1	If so, what?	Several months of integration using Mule ESB	Some experience with Mule ESB (worked with it for approxametly 6 weeks)	Some experience of flows and also deploying the ESB on server	I have participated in integration of 2 nodes with Mule ESB	I have connected multiple nodes to the same ESB so that they don't need to know more about each other than that the ESB exists and takes care of everything. Transformation and connections if my main area of expertice.	I have some experience with software integration using Open ESB. I have been part of a team developing the integration logic for a project where many separate nodes had to communicate with each other.	I have been apart of project where a integration bus was used and i did some of the flows in the ESB. The reason we used a ESB was that we had in our software suit multiple systems that had to communicate with each other. The best way to achieve the communication in a structured manner was by using an ESB.
2	**Is it hard to decide which part of the system functionality really belongs in the ESB?**	Yes	Not in our project	Yes	No	Yes and no	No	Yes
2.1	If so, please elaborate.	There have been disscusions where functionality have been added to a node which are more suited for to a operation, instead of using existing functionality and enriching and/or removing uneccesary data in the ESB.	-	It's hard to get knowledge about it, it messy.	-	Basic functionality such as transformation and mediation seems like a "must have" to be able to call it a ESB, at the same time a twitter module, for instance, feels like a highly disposable function that indeed kan be nice to have, but nothing that should be placed within the scope of a ESB. I feel that, as of now, there is no clear standard that states what a ESB should be able to do		There have been some discussions whether to use existing functionality and aggregate and enrich the message or instead make more tailor suited functionality for each operations. We decided to use both methods.

2.2	If not, how do you decide?	-	In our project, we haven't allowed any direct communication between isolated nodes. All inter-node communication goes through the ESB	-	ESB is in charge of "traffic control" between nodes. System specific calculations is to be done in the nodes. So the decision of what is to be done in the ESB is "easy"	-	The API in each nodes where extensive so not much new functionality had to be added. Because of the functionality that was needed were already implemented.	-
3	**Do you experience an issue with slower communication between nodes because of the extra steps that comes with a ESB?**	I have not performed any benchmark tests	No	No	No	No	Yes	Yes
3.1	If so, please elaborate.	There is a lot of XSLT in our ESB which may degrade the performance	-	-	-		Most of our nodes are written in java so they are natively able to communicate with each other, so much of the protocol translation that are being done in the ESB could also as easily have been done in the separate nodes. The translation overhead will probably slow down the performance of the system.	We have allot of costly transformation in our ESB for example xml to object, object to xml and various xml parsing.
3.2	If not, is it because the extra steps do not take any time or because the system does not require response?	-	I haven't seen any actual performance comparisons, but Mule ESB appears to be a fairly fast ESB implementation. Responses are required in most of the flows.	You hardly notice any difference in performance	I have only been using simple and fast function calls through the ESB. These calls have not been enough to cause any time delay.	This depends on what is done in the ESB but in general it should not be an issue as the ESB only performs tasks that otherwise would have to be done by other nodes in the system.	-	-
4	**Have you ever migrated older, integrated systems to the ESB architecture?**	No	No	No	No	No	No	No
4.1	If so, please describe the main issues encountered.	-	-	-	-	-	-	-

4.2	What did you do to resolve said issues?	-	-	-	-	-	-	-
5	**Are you able to get support or find information to resolve the issues that are encountered during integration?**	We have had problems getting support and documentation	This can be difficult...	No	Yes	Yes and no	The documentation is a bit lacking and can be confusing.	There has been some problems getting information out of the documentation. We have had problems getting documentation for the latest version of the ESB the older versions are not compatible with the current version. Also much of the examples that exists does not work or are incomplete.
5.1	If so, how do you find/get said help?	We have mainly been reusing something we have gotten to work previously, not knowing if it is the correct method.	-	-	From forums and some help from experts. As work progresses, more and more knowledge is achieved by "trial and error"	If you get an enterprise license (or if your opinion of the product is really important), you might get in contact with the creators of the ESB. This expertis is superior to any available documentation, but not easy to obtain.	I usually just try till i have found the solution to a problem and if the same problem come up again i have documented how to solve it.	-
5.2	If not, how do you handle issues?	-	Especially when working with Mule ESB, because the documentation is severely lacking. To resolve these issues, we've had some contact with a company called Entiros and the developers of Mule, Mulesoft	Try to find information in the documentation or find somebody who can help me	-	Documentation on ESBs in general is scarce at best. If you don't fall into the the category mentioned above and hit an obstacle you either find another way to do it or create your own quick and dirty solution.	-	We had to guess much and if something worked we reused that.

6	**Have you encountered any issues, apart from the ones mentioned in this questionaire, when working with ESB integration?**	If the team developing the ESB is slow, other teams may be held up	-	No	Big changes between different versions of the ESB makes it hard when documentation isn't up to date	It can be hard to get a grip on how to get the most of the ESB (lack of "best practises"). Another confusing thing is when the ESB provides multiple environments. As with Mule for example, they have Mule standalone, MuleStudio and an alternative to build with maven. There are several functions that work with one or two of these version but not with the third. This creates even more confusion as documentation and community doesn't always clearly state which version is handled.	I have encountered problems when nodes that the ESB communicates with are not stateless meaning that they have a session id. ESBs are usually stateless so that they will easily scale over multiple servers without having to share sessions between them. The problem arise when a node require a session id to perform a operation. Do the ESB store sessions (reducing scalability), returning the session id to the calling node (reducing transparency) or performing a login and logout for each request (reducing performance by increasing communication overhead).	Much of the programming is done in XML. XML is not designed for use as a programming language and because of that it becomes hard the get an overview of the code. The ESB we used had an enterprise license cost so our budget became higher.
6.1	Have you been able to resolve any of these issues?	-		-	Yes	No, not really, as this is a issue for the creators of the ESBs	Yes	Yes

6.2	If so, please describe your solution.	-	-	-	Trial and error	The ESBs gets better and better all the time, along with that the creators need to step up in order to keep users. If documentation is not up to date, developers will either chose another alternative or flood support and forums.	We performed a login/logout for each request.	There are GUI editors to aid in creating this kind of XML code. But they are not bug free and does not display all options that are possible to do in the XML. So a combination of this methods were used.