Architecture Documentation Review

Chiel Peters, Omar Pakker, Mary Gouseti, Cindy Berghuizen

October 5, 2013

1 Introduction

In this document the architectural solution of the FlyWithUs airline reputation management system is reviewed. The review uses question sets and scenario analysis to determine the completeness of the architecture. Section two reviews the architecture using premade question sets from [2]. In section three multiple scenario's are analysed and how the architecture responds/copes with those scenario's.

2 Question sets

This section contains the answers to the question sets defined in [2]. Two question sets where mandetory for this review and the last was spefically taken in order to explain certain limitations in the current version of the architecture. The first subsection contains questions related to stakeholders and their concerns. In the second subsection the requirements and key design decisions are discussed. Third and final, the viewpoints in the architecture are reviewed.

2.1 Question set : Capturing the right stakeholders and concerns

Stakeholder	**1. State your stakeholder role. List the set of concerns you have that pertain to the architecture whose AD is being reviewed.**
EU Claim	As EU Claim there are two major concerns with the FlyWithUs system. One is that the privacy of the users must be guaranteed. The second is that the system is fair and transparent. No airline should have more influence than another.
AirFrance-KLM	AirFrance-KLM wants to be able to contact the user based on a review. The customer service wants detailed statistics and reports with regards to both external and internal reviews. Also bad reviews caused by factors outside of the hands of the Airline companies should not affect the overall rating, hence cross-matching bad weather with flight information should be possible. Other concerns are: costs and customer satisfaction.
Dutch Government	As a representative of the Dutch government, Privacy and GreenIT are the main concerns in this project.
Initiator	The project must become successful in that the FlyWithUs system becomes the number one airline review website of the web. The main concerns for the Initiator are the overall functionality of the system, profitability and time to market.
Stakeholder	**2. Find and record all places in the AD where your stakeholder role is listed as being covered.**

EU Claim	Appendix A, page 16, Stakeholder 2: Sven-Erik Haitjema.
AirFrance-KLM	Appendix A, page 16, Stakeholder 4: Sinan Ceylan.
Dutch Government	Appendix A, page 16, Stakeholder 3: Philipp Darkow.
Initiator	Appendix A, page 16, Stakeholder 1: Peter Klijn.
Stakeholder	**3. Find and record all places in the AD where your concerns are listed as being addressed.**
EU Claim	Page 17: R1 Page 19: Q4, Q5
AirFrance-KLM	Page 16: F2 Page 18: M3 Functional Requirements: 5, 8, 9 Page 19: Q2
Dutch Government	Page 6: GreenIT [R2] Page 17: R1 Page 17: R2
Initiator	Page 6: Vendor lock-in [M2] Page 16: F1 Page 17: G1 Page 17/18: M1 Page 18: M2 Functional Requirements: 1 t/m 9 Page 19: Q1 t/m Q5
Stakeholder	**6. Record all concerns you have that are not listed as being covered in either the AD or any framework being used or that are listed in an unclear fashion. For each, state the impact of this omission or misunderstanding on project success.**
EU Claim	Both of EU-Claim's concerns are mentioned briefly, but stated in an unclear manner. Transparency/fairness are mentioned in Q4 however nowhere is mentioned how the system filters the reviews in order to keep the system fair. Privacy is mentioned as logging in and obtaining a certain role. But how privacy works in regards with the all overseeing administrator is not clear. This could become vital for project success if privacy is not well governed (Reputation).
AirFrance-KLM	All of AirFrance-KLM concerns are stated except cross-matching the reviews with flight information/weather data. Although not critically important this feature does help the airlines.

Dutch Government	Currently the same concern as EU Claim is not covered; how privacy is handled within the system. Are reviews posted with all public information of the user or anonymously? Who can access user data?
Initiator	As Initiator, the business concerns are clearly covered within the document however the functionality still is unclear. The functional requirements are mentioned however the viewpoints do not always make clear how the functionality will be achieved.
Stakeholder	**7. For each of your concerns as a stakeholder, find and record the places in the AD where that concern is addressed (not just listed). Explain why you do or do not believe that the concern will be satisfied by the architecture.**
EU Claim	Privacy: Addressed on page 10 at Airline Rating Service Database on page 25 at separate users table. The concern related to where the information is placed is clearly stated however how privacy is handled related to reviews and administrators is unclear. The user database will be protected and located under Dutch law. Transparency/Fairness: It is touched upon in page 15 at 'Filter and Store' and at 'Extract and Apply'. However the business rules related to filtering and fairness are not discussed and hence the architecture does not properly satisfy this concern.
AirFrance-KLM	Usability: Addressed on page 6 at "fast and easy to query"; page 6 at "performance"; page 8 at "B2C application"; page 10 at "Airline Rating Service database"; Section 2.2; page 18 at "functional requirements"; page 19 at "Quality attribute 2". The functional requirements related to AirFrance-KLM are stated in multiple places and this should suffice for the architecture. Customer contact: Mentioned on page 19 in the requirements but not in the viewpoints. Therefore the architecture does not properly satisfy this concern.
Dutch Government	GreenIT: Addressed on page 6 at "GreenIT [R2]" GreenIT is mentioned in the business view. However, it does not give any grounded argumentation or facts for the choice of using a non-relational database. It seems to mostly rely on guesswork and a 'gut-feeling'. The security/privacy concerns related to the location of the database are clearly stated and the architecture will be able to satisfy this concern. The privacy within the system itself is not as this is not addressed in any section or viewpoint.
Initiator	Functional requirements addressed on page 18. The functional requirements are listed but are not explained in detail through any of the viewpoints. The API in the B2C and B2B viewpoints addresses some of these concerns but the architecture needs some refinement to satisfy these concerns.
Stakeholder	**8. Find and record the place in the AD that prioritizes the concerns. Explain why you do or do not agree with it.**
All	There is no prioritization of the concerns. Some are mentioned in a way that they "must" or "should" be implemented, which may indicate a prioritization. The document in general lays a high focus on how the data is handled which implies a priority to performance and scalability.
Stakeholder	**9. Record important stakeholders that you are aware of that are not listed and whose concerns are not represented in the AD.**
All	All the stakeholders are listed.

Stakeholder	**10. State how you know that the architecture satisfies the concerns of the missing stakeholders and where this information can be found in the AD.**
All	There are no missing stakeholders.

2.2 Question set : Identify architecturally significant requirements and key design decisions

1. Are specific architecturally significant requirements (i.e., the sub-set of functional, quality attribute, and business requirements that shape the architecture under consideration) identified?
The requirements are stated within the Appendix A. They are divided into three subcategories:

1. Business Goals
2. Functional requirements
3. Quality attributes

The business goals derive specific requirements related to financial objecives, growth, social responsibility and market position. The business goals are stated using the template in[1].
The functional requirements state the functionality that the system must support. On key requirement of the FlyWithUs system that is not mentioned is : messaging system. Currently the requirements state:
Business to Business clients need to be able to respond to posts done by Business to Consumer users via the FlyWithUs application
However this does not capture that the conversation should occur privately between the business and the clients. This would change the architecture as privacy should be considered as a main concern.
The quality attributes state the main quality attributes and how they apply to the system and stakeholders. EU-Claim called for performance however this is not mentioned elsewhere in the document. Privacy and security are taken as one while it is not mentioned how privacy applies towards the system.

2. Are ASRs represented in a clear, unambiguous manner (c.f., 6-part quality attribute scenarios [Bass 2003])? Is the utility of the requirements documented in terms of what the system does and how it meets the customers expectations?
At first, ASRs are introduced and related to the stakeholders. Then they are expressed as business goals which clarifies these ASRs; however, business goal M3 (*Improve KLM airliner quality in comparison with other*) implies that KLM will be dealt differently than the other airlines while a requirement was to deal with all the airlines in the same way. After the business goals the functional requirements are stated. Functional requirements are expressed as a list of things the system must/should do. However, it is not clear if this vocabulary is used to prioritize the requirements. Additionally the services the system provides to users (business or clients) are briefly mentioned and not elaborated which makes them unclear, for example it is mentioned that the data must be fair but it is not explained what that means and how it affects architecture. As a result the system's functionality is not clearly documented which makes it difficult to conclude if it meets customer's expectations.

3. Are there remaining requirements that could come up later and have a significant impact on the architecture? How will the architecture (and the architecting process) react to the emergence of new ASRs?
One big assumption mentioned in the beginning of the document is that the data is not big data. Considering the growth of social networks this may become a requirement in the near future. The current model is a pipeline model which does not scale with the amount of incoming external resources. Therefore a large part of the ETL module would be required to change in order to cope with the big data.

4. Is the relationship between ASRs documented and understood (e.g. between performance requirements in a distributed system and the bandwidth reliability and stability of the supporting network transport systems)?
The relation between scalability and performance is documented and can easily be understood. Other than that relation no relationships are mentioned in the requirements.

5.Are decisions represented in a clear, unambiguous manner (c.f., architectural tactics [Bass 2003])? Is the rationale for key decisions captured? Are the costs and resources associated with implementing the decisions documented? The main decisions are presented using the template from []. Apart from the group field that is not correctly used because performance is included in this field, the remaining decisions are clear and unambiguous. The rational of every decision is explained in the arguement field. Costs and resources are mentioned but not thoroughly explained.

6. Are there remaining architectural decisions or impacts (e.g., issues or problems that could come up during deployment, deferred decisions that need to be bound later)?
Currently for both the B2C and B2B modules all functionality is mentioned as API. There are decisions related to how this API functions what interactions there are possible between the user and its API and how the API obtaines the data from the storage.

7. Is there a mapping between decisions and requirements? The format used is defined in [2]. The table contains the row related requirements which does this mapping. However it is not followed as strictly as elsewhere in the document where the requirement code (G1, R1 etc.) is used.

11. Are specific driving architectural decisions identified? Is the relationship between them documented and understood (e.g., between performance requirements in a distributed system and the bandwidth, reliability, and stability of the supporting network trans-port systems)?
Most design decisions are of architectural kind and may not be comprehensable to the stakeholders. The format used is defined in [2] and contains a row related design decision which indicates with decisions are related. How they are related is however not mentioned.

12. Are decisions represented in a clear, unambiguous manner? Is the rationale for key decisions captured? Are the costs and resources associated with implementing the decisions documented? Arguments and implications are provided for most of the main decisions which makes their significance clear to the stakeholders. However, the costs are not analysed enough.

13. Are there remaining architectural decisions or impacts (e.g., issues or problems that could come up during deployment, deferred de-cisions that need to be bound later)?
As mentioned towards the architects there are remaining design decisions related towards the functionality of the B2B and B2C modules and how these modules are communicating.

14. Do you understand how the AD will identify constraints and implementation responsibilities (e.g. delegated decisions?)
The constraints are given in the template of [1] in the row constraints however the responsibilities are not delegated towards the stakeholders. This is because only the stakeholders that use the system are mentioned and not the stakeholders that will implement this system.

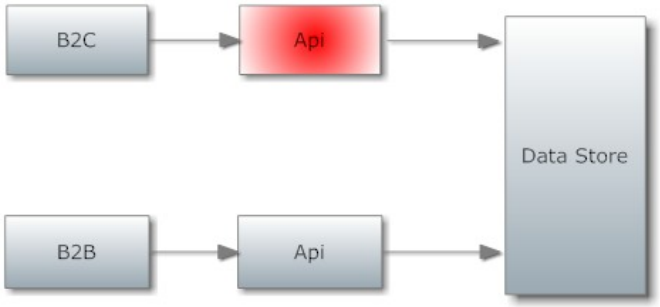
2.3 Question set : Reviewing the choice of Viewpoints

3 Scenarios

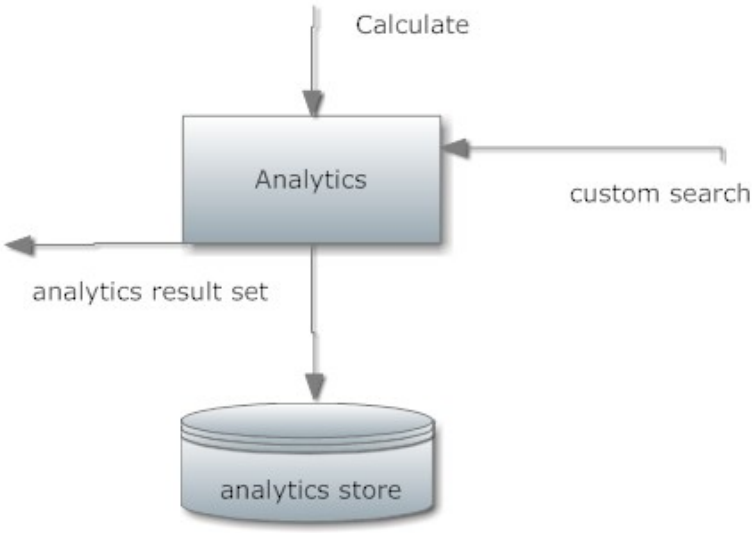
This section contains four scenarios and how the architecture responds to them. The format is obtained from [1]. The choices for the scenarios were made on multiple factors:

1. To demonstrate one of the key design decisions and the related trade-offs and sensitivity points (3.1)
2. To indicate that some (implicit) design decisions are still open for discussion (3.2,3.3)
3. To test how the architecture responds to a change in an assumption and which components need to be modified (3.4)

3.1 Scenario: Failure in B2B/B2C

<table>
<tr><td>Scenario</td><td>B2B - B2C decoupling</td></tr>
<tr><td>Attribute</td><td>Availability</td></tr>
<tr><td>Environment</td><td>Normal Operations</td></tr>
<tr><td>Stimulus</td><td>B2B or B2C failure</td></tr>
<tr><td>Response</td><td>Failure of either one does not influence the other.</td></tr>
<tr><td></td><td>
<table>
<tr><th>Architectural Decision</th><th>Sensitivity</th><th>Trade-off</th><th>Risk</th><th>Non-risk</th></tr>
<tr><td>Decoupling</td><td>S1</td><td>T1</td><td></td><td></td></tr>
<tr><td>Different API</td><td>S1</td><td>T2</td><td></td><td>N1</td></tr>
</table>
T1: When both systems are decoupled and both access the same data store, a trade-off is created. The performance of the B2C clients is directly linked to the performance of the B2B clients.
T2: The API contains duplicate functionality hence redundant code might exist. This is a trade-off between modifiability/maintainability because of the redundant code vs availability (if one system crashes, the other is not influenced).
N1: If the API was not split up, there would be a single point of failure, as mentioned by the architects.
S1: If the B2B and B2C systems are running on the same server and the server goes down, B2C and B2B will both go down.
</td></tr>
<tr><td>Reasoning</td><td>Decoupling ensures, together with the different API, that if either the B2B or B2C module goes down, the other stays unaffected.</td></tr>
<tr><td>Architecture Diagram</td><td>The red API crashes.
 
</td></tr>
</table>

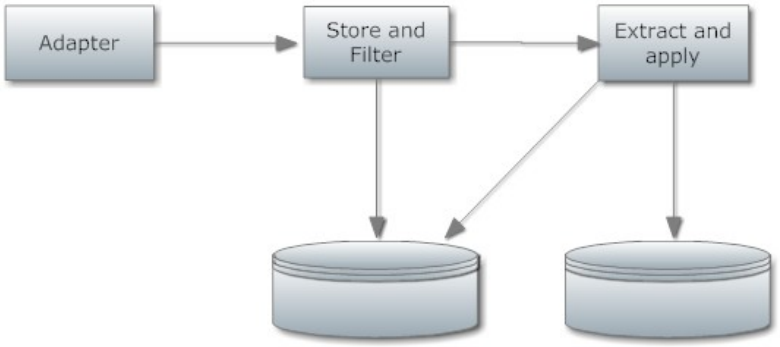
3.2 Scenario: Peak-load on the analytics module

<table>
<tr><td>Scenario</td><td>Peak-load on the analytics module</td></tr>
<tr><td>Attribute</td><td>Performance</td></tr>
<tr><td>Environment</td><td>Normal Operations</td></tr>
<tr><td>Stimulus</td><td>Recalculating ratings/reviews</td></tr>
<tr><td>Response</td><td>B2B latency ¡10 sec.</td></tr>
<tr><td></td><td>
<table>
<tr><th>Architectural Decision</th><th>Sensitivity</th><th>Trade-off</th><th>Risk</th><th>Non-risk</th></tr>
<tr><td>Recalculate 1/2 times a day (Implicit)</td><td></td><td></td><td>R1</td><td></td></tr>
<tr><td>B2B accesses analytics directly (Implicit)</td><td></td><td>T1</td><td></td><td></td></tr>
</table>
R1: Recalculating the final rating once or twice a day creates a peak-load situation on the analytics module and might lead to slow or unresponsive connections between the analytics module and the B2B clients.
T1: B2B clients can directly access the analytics module, which gives the B2B clients the freedom to perform custom searches. The trade-off of this is that when the module is overloaded, the functionality may be unavailable.
</td></tr>
<tr><td>Reasoning</td><td>Recalculating on a time frame creates a peak-load situation which may lead to an undesirable response time when a B2B user tries to perform a custom search.</td></tr>
<tr><td>Architecture Diagram</td><td>Analytics module overloaded.
 
</td></tr>
</table>

3.3 Scenario: Review privacy

<table>
<tr><td>Scenario</td><td>Review Privacy</td></tr>
<tr><td>Attribute</td><td>Privacy</td></tr>
<tr><td>Environment</td><td>Normal Operations</td></tr>
<tr><td>Stimulus</td><td>User writes a review</td></tr>
<tr><td>Response</td><td>Anonymous review</td></tr>
<tr><td></td><td>
<table>
<tr><th>Architectural Decision</th><th>Sensitivity</th><th>Trade-off</th><th>Risk</th><th>Non-risk</th></tr>
<tr><td>Logged In</td><td></td><td></td><td>R1</td><td></td></tr>
</table>
R1: The anonymity of the user posting a review is not guaranteed.
</td></tr>
<tr><td>Reasoning</td><td>The architecture does not state how anonymity of reviews is guaranteed, assuming that a review will be shown anonymous.</td></tr>
<tr><td>Architecture Diagram</td><td>Anyone can see review details.
 
</td></tr>
</table>

3.4 Scenario: Big Data

<table>
<tr><td>Scenario</td><td>Big Data</td></tr>
<tr><td>Attribute</td><td>Performance, Availability</td></tr>
<tr><td>Environment</td><td>Normal Operations</td></tr>
<tr><td>Stimulus</td><td>Big Data input</td></tr>
<tr><td>Response</td><td>Handle all input without any loss of data.</td></tr>
<tr><td></td><td>
<table>
<tr><th>Architectural Decision</th><th>Sensitivity</th><th>Trade-off</th><th>Risk</th><th>Non-risk</th></tr>
<tr><td>ETL Adapters</td><td></td><td></td><td></td><td>N1</td></tr>
<tr><td>ETL Approaches</td><td>S1</td><td></td><td></td><td></td></tr>
<tr><td>Pipeline Structure</td><td></td><td></td><td>R1</td><td></td></tr>
</table>
S1: If the amount of external input grows and becomes Big Data, the system will be unable to handle the input due to the pipe model.
N1: The adapters in the ETL are independent and therefore easily modifiable.
R1: The Filter and Store and Extract and apply module process all reviews sequentially which is a risk regarding scalability.
</td></tr>
<tr><td>Reasoning</td><td>The pipeline structure won't be able to process large sets of data (Big Data) as input and won't scale easily.</td></tr>
<tr><td>Architecture Diagram.</td><td> 
</td></tr>
<tr><td>Notes</td><td>The architectures note that a dramatically higher number of reviews would choke the system. It was therefore an explicit choice to not handle big data, however this scenario is valuable in that it identifies the components that need to be modified in order to make it more scalable.</td></tr>
</table>

References

[1] Bass et al. *Software Architecture in Practice*. Addison Wesley, Boston, 3rd Edition, 2012.

[2] Tyree, J. Akerman, A. *Architecture Decisions: Demystifying Architecture*, IEEE Software, 2005