

# 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 were mandatory for this review and the last was specifically 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      | <b>1. State your stakeholder role. List the set of concerns you have that pertain to the architecture whose AD is being reviewed.</b>  |
|------------------|--|
| 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 outside of the blame of the Airline companies should not affect the overall rating, hence crossmatching 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. My main concerns are the overall functionality of the system, profitability and time to market.  |
| Stakeholder      | <b>2. Find and record all places in the AD where your stakeholder role is listed as being covered.</b>   |

|                    |   |
|--------------------|---|
| 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.  |
| <b>Stakeholder</b> | <b>3. Find and record all places in the AD where your concerns are listed as being addressed.</b>   |
| EU Claim           | Page 17: R1<br>Page 19: Q4, Q5  |
| AirFrance-KLM      | Page 16: F2<br>Page 18: M3<br>Functional Requirements: 5, 8, 9<br>Page 19: Q2   |
| Dutch Government   | Page 6: GreenIT [R2]<br>Page 17: R1<br>Page 17: R2  |
| Initiator          | Page 6: Vendor lock-in [M2]<br>Page 16: F1<br>Page 17: G1<br>Page 17/18: M1<br>Page 18: M2<br>Functional Requirements: 1 t/m 9<br>Page 19: Q1 t/m Q5  |
| <b>Stakeholder</b> | <b>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.</b>  |
| 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 is privacy handled within the system. Are reviews posted with all public information of the user or anonymously?   |
| Initiator        | As Initiator, the business concerns are clearly covered within the document however the functionality still is unclear. The functional requirements are mentioned however it is not always clear how these will be achieved in the viewpoints.  |
| Stakeholder      | <b>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.</b>  |
| EU Claim         | <p>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 is unclear. The user database will be protected and located under dutch law.</p> <p>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 it is hard for the architecture to satisfy this concern.</p> |
| AirFrance-KLM    | <p>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".</p> <p>The functional requirements related to AirFrance-KLM are stated in multiple places and this should suffice for the architecture.</p> <p>Customer contact: Mentioned on page 19 in the requirements however not in the viewpoints. Therefore the architecture will probably not satisfy this concern.</p>             |
| Dutch Government | <p>GreenIT: Addressed on page at "GreenIT [R2]"</p> <p>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 gut-feeling. The privacy concerns related to the location of the database are clearly stated and the architecture will be able to satisfy this concern. The privacy in the system however is not addressed.</p>   |
| Initiator        | <p>Functional requirements addressed on page 18.</p> <p>The functional requirements are not stated very clear and most of them are not explained in detail through out the viewpoints. The API in the B2C and B2B viewpoints must address most of these concerns. However the architecture must be defined more in detail to satisfy these concerns.</p>  |
| Stakeholder      | <b>8. Find and record the place in the AD that prioritizes the concerns. Explain why you do or do not agree with it.</b>  |
| 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 gives a priority to performance and scalability.   |
| Stakeholder      | <b>9. Record important stakeholders that you are aware of that are not listed and whose concerns are not represented in the AD.</b>   |
| All              | All the stakeholders are listed.  |

|             |   |
|-------------|---|
| Stakeholder | <b>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.</b> |
| 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 objectives, 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 rationale of every decision is explained in the argument 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 obtains 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 transport systems)?**

Most design decisions are of architectural kind and may not be comprehensible 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 decisions 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.

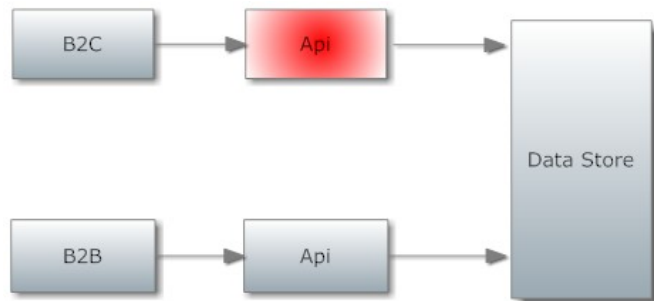
## 2.3 Question set : Reviewing the choice of Viewpoints

### 3 Scenarios

This section contains four scenario's and how the architecture responds to them. The format is obtained from [1]. The choices for scenario's 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


| Scenario               | B2B - B2C decoupling   |          |      |          |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
|------------------------|--|----------|------|----------|--|------------------------|-------------|----------|------|----------|------------|----|----|--|--|---------------|----|----|--|----|
| Attribute              | Availability   |          |      |          |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
| Environment            | Normal Operations  |          |      |          |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
| Stimulus               | B2B or B2C failure   |          |      |          |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
| Response               | Failure of either one does not influence the other.  |          |      |          |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
|                        | <table><thead><tr><th>Architectural Decision</th><th>Sensitivity</th><th>Tradeoff</th><th>Risk</th><th>Non-risk</th></tr></thead><tbody><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></tbody></table> <p>T1: When both systems are decoupled and both access the same data store a tradoff is created. The performance of the B2C clients is directly linked to the performance of the B2B clients</p> <p>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 (one system crashes does not influence the other system).</p> <p>N1: If the API was not split up as well then this would create a single point of failure as mentioned by the architects.</p> <p>S1: If the B2B and B2C systems are in the same server and the server goes down, B2C and B2B will both go down.</p> |          |      |          |  | Architectural Decision | Sensitivity | Tradeoff | Risk | Non-risk | Decoupling | S1 | T1 |  |  | Different API | S1 | T2 |  | N1 |
| Architectural Decision | Sensitivity  | Tradeoff | Risk | Non-risk |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
| Decoupling             | S1   | T1       |      |          |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
| Different API          | S1   | T2       |      | N1       |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
| Reasoning              | Decoupling ensures, together with the different API, that if one the B2B or B2C module goes down, the other won't.   |          |      |          |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |
| Architecture Diagram   | <p>The red API crashes.</p>  <pre>graph LR; B2C[B2C] --&gt; Api1[Api]; B2B[B2B] --&gt; Api2[Api]; Api1 --&gt; DS[Data Store]; Api2 --&gt; DS; style Api1 fill:#ff0000; style Api2 fill:#0000ff;</pre>  |          |      |          |  |                        |             |          |      |          |            |    |    |  |  |               |    |    |  |    |



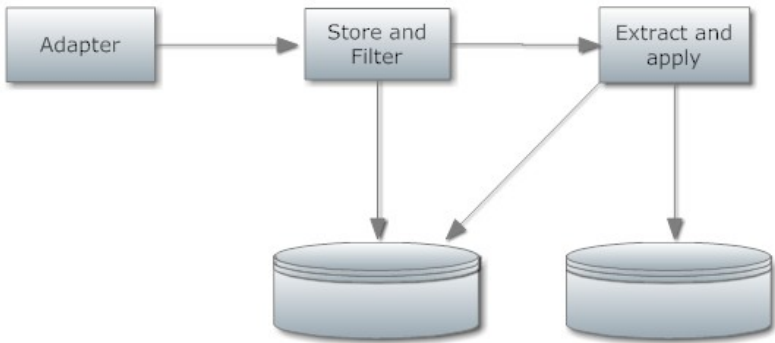
### 3.2 Scenario: Peak load on analytics module

| Scenario   | Peak load on the analytics module  |             |          |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
|--|--|-------------|----------|------|----------|------------------------|-------------|----------|------|----------|--|--|--|----|--|--|--|----|--|--|
| Attribute  | Performance  |             |          |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
| Environment  | Normal Operations  |             |          |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
| Stimulus   | Recalculating ratings/reviews  |             |          |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
| Response   | B2B latency $\leq 10$ sec.   |             |          |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
|  | <table><tr><th>Architectural Decision</th><th>Sensitivity</th><th>Tradeoff</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> |             |          |      |          | Architectural Decision | Sensitivity | Tradeoff | Risk | Non-risk | Recalculate 1/2 times a day (Implicit) |  |  | R1 |  | B2B accesses analytics directly (Implicit) |  | T1 |  |  |
|  | Architectural Decision   | Sensitivity | Tradeoff | Risk | Non-risk |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
|  | Recalculate 1/2 times a day (Implicit)   |             |          | R1   |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
|  | B2B accesses analytics directly (Implicit)   |             | T1       |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
| <p>R1: Recalculating the final rating once or twice a day creates a major peak load on the analytics module and might lead to an unavailable analytics module for the B2B clients.</p> <p>T1: B2B clients can directly access the analytics module, this gives the B2B clients the freedom to perform custom searches, but also leads to unavailability if the module is overburdened.</p> |  |             |          |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
| Reasoning  | Recalculating on a time frame creates a massive peakload which may lead to an undesirable response time when the B2B performs a custom search.   |             |          |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |
| Architecture Diagram   | <p>Analytics module overloaded.</p> <pre>graph TD; Calculate[Calculate] --&gt; Analytics[Analytics]; CustomSearch[custom search] --&gt; Analytics; Analytics -- "analytics result set" --&gt; ResultSet[analytics result set]; Analytics --&gt; AnalyticsStore[(analytics store)];</pre>                                   |             |          |      |          |                        |             |          |      |          |  |  |  |    |  |  |  |    |  |  |

### 3.3 Scenario: Review privacy

| Scenario               | Review Privacy   |          |      |          |  |                        |             |          |      |          |           |  |  |    |  |
|------------------------|--|----------|------|----------|--|------------------------|-------------|----------|------|----------|-----------|--|--|----|--|
| Attribute              | Privacy  |          |      |          |  |                        |             |          |      |          |           |  |  |    |  |
| Environment            | Normal Operations  |          |      |          |  |                        |             |          |      |          |           |  |  |    |  |
| Stimulus               | User writes a review   |          |      |          |  |                        |             |          |      |          |           |  |  |    |  |
| Response               | Anonymous review   |          |      |          |  |                        |             |          |      |          |           |  |  |    |  |
|                        | <table><tr><th>Architectural Decision</th><th>Sensitivity</th><th>Tradeoff</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> <p>R1: The anonymity of the user posting a review is not guaranteed.</p> |          |      |          |  | Architectural Decision | Sensitivity | Tradeoff | Risk | Non-risk | Logged In |  |  | R1 |  |
| Architectural Decision | Sensitivity  | Tradeoff | Risk | Non-risk |  |                        |             |          |      |          |           |  |  |    |  |
| Logged In              |  |          | R1   |          |  |                        |             |          |      |          |           |  |  |    |  |
| Reasoning              | In the architecture it is not stated that a review will be shown anonymous or how anonimity of reviews is guaranteed.  |          |      |          |  |                        |             |          |      |          |           |  |  |    |  |
| Architecture Diagram   | <p>Anyone can see review details.</p>  <pre>graph LR; user((user)) -- "Writes review" --&gt; reviews[(reviews)]; show_review((show review)) --&gt; reviews;</pre>                  |          |      |          |  |                        |             |          |      |          |           |  |  |    |  |

### 3.4 Scenario: Big data

|                              |   |                    |                 |             |                 |
|------------------------------|---|--------------------|-----------------|-------------|-----------------|
| <b>Scenario</b>              | Big Data  |                    |                 |             |                 |
| <b>Attribute</b>             | Performance, Availability   |                    |                 |             |                 |
| <b>Environment</b>           | Normal Operations   |                    |                 |             |                 |
| <b>Stimulus</b>              | Big Data input  |                    |                 |             |                 |
| <b>Response</b>              | Handle all input without any loss of data.  |                    |                 |             |                 |
|                              | <b>Architectural Decision</b>   | <b>Sensitivity</b> | <b>Tradeoff</b> | <b>Risk</b> | <b>Non-risk</b> |
|                              | ETL Adapters  |                    |                 |             | N1              |
|                              | ETL Approaches  | S1                 |                 |             |                 |
|                              | Pipeline Structure  |                    |                 | R1          |                 |
|                              | <p>S1: If it is Big data the system can not handle the input a choke point will occur in the pipe model.</p> <p>N1: The modules in the ETL are independent and therefore easily modifiable.</p> <p>R1: The <i>Filter and Store</i> and <i>Extract and apply</i> module process all reviews sequentially which is a risk regarding scalability</p> |                    |                 |             |                 |
| <b>Reasoning</b>             | The pipeline structure won't be able to process large input sets of data (Big data).  |                    |                 |             |                 |
| <b>Architecture Diagram.</b> |  <pre> graph LR     Adapter[Adapter] --&gt; SF[Store and Filter]     SF --&gt; EA[Extract and apply]     SF --&gt; DS1[(Data Store 1)]     EA --&gt; DS2[(Data Store 2)] </pre>   |                    |                 |             |                 |
| <b>Notes</b>                 | <p>The architectures note that <i>a dramitacally higher number of reviews would choke the system</i>. 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.</p>  |                    |                 |             |                 |

## 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