Dear reviewers and editors

Please find enclosed the updated version of our paper: "Ten years of Stream Reasoning. Now what?".

We carefully took your comments into account: they inspired useful discussions among us and ended up in the following set of actions that we took to address them. In the following, we first describe the main actions we took based on the meta-review; then we comment in detail on the points raised in the reviews.

The first point of the meta-review states:

> *You are mainly considering work from the field of the Semantic Web (reviewer 1): This is in my view the most serious reported shortcoming (or, if you disagree, I would like to hear your opinion). This can be addressed by either including the missing parts of the literature, or by clarifying the scope of this work (such that the missing literature is out of scope).*

We cannot deny that our main background is this area and we recognise that the studies we analysed in Section 2 are among the main results obtained in this context. We took three actions to address this concern:

A1. In the introduction, we clarified the content of the paper. We explain that Section 2 is mainly an analysis of the results obtained so far by the semantic web research in stream reasoning. Section 3 presents the challenges in the stream reasoning domain in 2017, citing ongoing studies from different areas (e.g., SemWeb, AI, DBMS).
A2. We described the semantic web concepts we are using (e.g. RDF and SPARQL).
A3. We checked the papers suggested by reviewer 1 and we integrated part of them in the article.

As a result, the updated version of our article covers a wider set of studies and is more accessible to a boarder set of readers.

The second point states:

> *The conclusion section is a bit disappointing and not what you would expect from a position paper (reviewer 3). I think that describing some "concrete action points or emerging fields" and possibly including some bold predictions or general points of criticism would indeed strengthen the paper and increase its potential impact.*

We believe that this request is largely covered in Section 3. Action A1 partially addresses this issue. Moreover,

A4. We reworked the conclusion to point to the part of the paper, where the concrete actions and emerging fields are described.
A5. We moved the quoted sentence (the "idea") from the introduction to the conclusion, to re-set the final goal.

The third point states:

> *The definition of stream reasoning is missing, and the relation to stream processing is therefore unclear (reviewer 1). I imagine that this will be easy to fix, as you seem to build upon a precise notion of stream reasoning that is clearly different from mere stream processing, but this notion and this difference are nowhere made explicit. In fact, the quote on page 3 could serve as the basis for such a definition, but is currently only labeled an "idea".*

Thanks for pointing out this important point. Looking at the reviews and at the content, we realised that we misused the stream processing definition. We took three actions to address this point. One is Action A5 as described above, the others are:

A6. In the introduction, we clarified what we mean by stream processing (i.e. processing of data in motion) and stream reasoning (i.e. reasoning over flows of data)
A7. We revised the content of the paper to use the "stream reasoning" and "stream processing" according to those definitions in A6. Where it was possible we also used more accurate terms such as DSMS and CEP.

The last point is:

> *Furthermore, I agree with reviewer 1 that it would be beneficial if the informal requirements can be made more formal. I see, however, that this might be very difficult to achieve in a general manner that applies to all possible kinds of stream reasoning and that doesn't favor any particular application domain. These informal requirements are certainly useful, but they can probably be made more clear-cut to avoid the type of confusions that reviewer 1 reports.*

While we recognise the importance of formal requirements in designing systems and solutions (that must provide features to satisfy them), we do not believe that research trends/areas can or must be described through formal requirements. We are not aware of any research areas and trends that provide formal requirements to say if something is part or not of it. What are the formal requirements to decide if a study is data science or not? And for Big Data processing? On the contrary, what we observed so far is that areas are usually defined in a generic way and it is very hard---if not impossible---to define their edges. Indeed, research fields are usually defined through a social process rather than a formal specification; see

- C. Banville and M. Landry: Can the Field of MIS be Disciplined. CACM 32(1): 48-60 (1989)

for an explanation. And we believe that it is something valuable, since these are spaces that offer opportunities to combine studies of different areas and to generate new results and studies. We, hence, did not take any action about the requirements: what is and what is not stream reasoning can be a valuable question, but the answer can hardly be given by looking at some formal requirement.

Moreover, we reviewed the existing work on benchmarking, as requested by reviewer 3. Please find attached the new document. Among all the changes, we highlighted in purple the major ones in Sections 2 and 3. We hope it will help in the review process.

We would like to thank you all for the effort you put in reviewing the article.

Reviewer #1

> *[...] In my view, it is relevant to distinguish between stream processing and stream reasoning. Stream processing is about doing simple transformations on streams like selection, projection, joins and mappings. The common characteristic is that the result is always a subset of the input or a straight forward mapping from input to output.*

We used to agree with this definition. After discussions and research, we realised that the meaning of this definition had evolved. Today stream processing is referred to a computation paradigm, typically opposite to batch processing, which studies and develop techniques to process data in motion. We do agree that we misused the term in the paper. To fix this, we explained our definition in the introduction (Action A6), and we carefully revised the occurrences of stream processing/stream processor with more accurate terms (Action A7).

> *Stream reasoning, on the other hand, is about inferring implicit or new information. No such distinction is made in paper.*

Given the actions above the distinction should be clearer now.

> *As a consequence, almost anything is stream reasoning. Is for example a UNIX shell supporting pipes a stream reasoner? Is the UNIX tool tr a stream reasoner (it takes a stream as input and replaces characters in it)? The definition that is used refer to "logical reasoning in real-time on [...] noisy data streams" and still the authors claim that only some stream reasoning approaches work in real-time and most do not consider noisy data at all. Does that mean that there are no stream reasoning tools in existence? I beg to differ. The fact that we lack a clear definition of stream reasoning is one of the major open problems. This is not addressed.*

We revised the introduction and we introduced a definition for stream reasoning.

About the quoted sentence of [37] (new version of the paper), we moved it in the conclusions and we made clear that we intend it as a goal rather than a definition of an "envisioned stream reasoner". We agree that so far such an envisioned stream reasoner lacks a functioning implementation. We do not believe, however, that this implies that there are no contributions to stream reasoning, as many works offer crucial, partial results to reach the overall goal. To make an analogy, would you say that research to bring people to Mars does not exist because no one has been to Mars? The fact that a goal has not been reached does not imply that no efforts have been done to achieve it. This quoted sentence is setting a goal, and efforts to achieve it are parts of stream reasoning (with obvious limits, e.g. studies on how to speed up networks are not part of stream reasoning, even if networks are needed to move streams). So what remains is the differentiation between important building blocks or stepping stones and things that are outside the consideration of the field. We believe that in accordance to Banville and Landry (1989) this is a task for the research community to determine – an opportunity to bring together ideas from different areas to address the common goal rather than a limit.

> *When it comes to the requirements, they are generally good and relevant, but they are also only informal. This makes it very hard to determine if a formalism satisfies the*

> *requirement or not. For example, take R8 Complex domains. I would argue that all SQL-based solutions, like most DSMSs, have quite good support for complex domains. Still the authors claim that this is not the case as seen in Table 1. What is the argumentation there?*

The main purpose of the table was to give an idea about the fact that there was a chance to put together solutions for different areas to create something better. In this sense, the cross (now empty space) did not mean that DSMS cannot handle very complex schema at all. We adjusted the caption of the table to clarify on this point.

> *When it comes to R3 Variety, I would argue that CEP are very good at handling variety as they usually consider a large set of different types of events and how these are combined into more complex events. To take a third example, R1 Volume. This seems obvious. If a system can handle millions of items then it must be able to handle volume. But what about a system that is scalable in the sense that it can run on an arbitrary number of processors, but it is extremely slow, would it be good at handling high volume data? It can scale to an infinite amount of data, if you have infinite amount of computational power. My conclusion is that the requirements make sense, but are quite hard to use in practice since they are too vague. Again, coming up with formal requirements that can actually be tested would be a major research contribution. The requirements could be in the form of specific use-cases.*

Thanks for pointing out the interpretational flexibility of some of our requirements. As mentioned we believe it is the task of the research community to take these requirements and show their different aspects (as you just did with scalability) and investigate their trade-offs. The requirements, which we proposed, want to show (in a non-exhaustive way) what stream reasoning should aim at targeting. They should be inspiring and push people to make/use research on stream reasoning. This is particularly true in a research area as fresh as this one (a more detailed answer can be found above in the AEs response).

> *Lastly, the paper is written by people in the area of semantic web mainly for other people in the semantic web. For example, RDF, RDF graphs and SPARQL are never explained.*

We added the references and brief explanations of those concepts in Section 2.

> *It is also the case that many results from classical AI areas such as belief revision, reasoning about action and change and runtime verification are not included.*

The editing actions described in detail below should have addressed this point.

> *S1a: In the introduction some examples of stream reasoning are provided. It is not clear to me whether all of them can currently be expressed in some existing stream reasoning formalism.*

We assume you refer to the questions in the first paragraph. Most of the questions have been studied in some stream reasoning paper (that is referenced). The last question, e.g., has been studied in [45] (new version of the paper) in a non-streaming context, but authors envision the use of stream reasoning in the future work section.

> *S2a: You define a landmark window as a window of infinite length containing the full stream. What does this definition give? Another name for a stream?*

While streams exist outside data stream management systems, windows do not. Windows are used by (some) engines to create access to the stream. The concept of landmark window is known in stream processing research, e.g. [30] (new version of the paper). Please note that we do not claim that the landmark window contains the full stream. On the contrary, it contains the stream from a time instant, which is not the time instant on which the stream starts to exist. For instance, if at 10:30 we created a landmark window over the Twitter stream, such a window would contain the tweets from 10:30 up to now, not the whole stream from the moment on which Twitter was created.

> *In my opinion a window should by definition be finite and a window should never be processed before it is complete. If it can be processed before, then no window is necessary.*

We agree on the fact that windows should be finite given a time instant, and all the windowing operators should create (finite) windows given a time instant. However, we do not agree on the fact that a window should be complete to be processed. Several real-world systems process the window content without considering if they are complete or not. This behaviour has been extensively studied in

- Nihal Dindar, Nesime Tatbul, Renée J. Miller, Laura M. Haas, Irina Botan: Modeling the execution semantics of stream processing engines with SECRET. VLDB J. 22(4): 421-446 (2013)
- Irina Botan, Roozbeh Derakhshan, Nihal Dindar, Laura M. Haas, Renée J. Miller, Nesime Tatbul: SECRET: A Model for Analysis of the Execution Semantics of Stream Processing Systems. PVLDB 3(1): 232-243 (2010)

as well as in [44] (new version of the paper).

> *S2b: I do not understand the following sentence "The graph-level entailment can be viewed as a direct application of SPARQL entailment regimes, since the inference process is taken into account in the context of the evaluation of graph patterns over graphs."*

The explanation of SPARQL and entailment regimes in Section 2 (Action A2) should clarify this sentence.

> *S2c: What do you mean by "applying the inference process context of the fixed windows"? I guess you mean it is computed based on the content of a window, but it is hard to understand.*

Thanks for pointing this out: we rephrased the sentence to make it easier to understand.

> *S2d: In 2.2 it sounds like the forgetting that is a natural part of windows is a problem. I would argue that the whole point of a window is to allow a user to define in what temporal context a statement should be evaluated. The semantics of the query is directly dependent on the window definition. If you want to reason over a longer*

> *horizon you either have to create a larger window or use a window-less method, like the logic-based stream reasoning approach [46].*

The idea behind 2.2 is that a user may want to operate on the data of the items captured in a window (not necessarily sliding), but also on the time instants. E.g., in a window of five minutes, identify the cases where the temperature in a turbine increases of 30% followed by a failure error message [102] (new version of the paper). These queries cannot be (trivially) modelled and answered by systems that do not use windows and time-aware operators on stream items.

> *S2e: The works on logic-based stream reasoning presented in Section 3.2.1 are also examples of stream level entailment. The work in [46] for example only process each stream element once and there is no need to store any previous elements as the necessary history is implicitly encoded in the formula.*

Action A1 sets the scope of Section 2 to studies in the semantic web area.

> *S3a: Regarding 3.1.1, there is work on spatio-temporal stream reasoning for example:*
> *Fredrik Heintz and Daniel de Leng. Spatio-Temporal Stream Reasoning with Incomplete Spatial Information. Proc. ECAI 2014.*
> *Daniel de Leng and Fredrik Heintz. Qualitative Spatio-Temporal Stream Reasoning With Unobservable Intertemporal Spatial Relations Using Landmarks. Proc. AAAI 2016.*

We added the reference and we rephrased the text accordingly.

> *S3b: I don't understand what separates "analytics-aware stream reasoning" from other forms of stream reasoning. It seems that the main thing is that they are based on a cloud infrastructure. The type of reasoning could be provided in other stream reasoning approaches as well. That said, analytics is definitely one of the main uses of stream reasoning (therefore most approach ought to be "analytics-aware").*

The goal was to indicating studies, where stream reasoning techniques are applied to real problems/contexts to make analytics. However, it looks that it created confusion, so we decided to rephrase and removing this name.

> *S3c: Is there a difference between an API and a query language? Both are ways of interacting with a system, one using a formal language the other function calls (and I guess in most cases sending a query is also a function call).*

As you pointed out, APIs and query languages are different. A query language is, in general, declarative and pursue the idea that the user should specify what she wants and not how it should be achieved. APIs offer flexibility in describing the task, in particular for complex tasks such as the ones in big data analytics. Most of the existing big data processors are studying declarative APIs to bring together the strong points of query languages and APIs, but what is the best way to "interact" with the system is still an open problem in the data management community.

> *S3d: Why do you consider all CEP-based stream reasoners to care about ontology axioms? Most CEP-engines don't. This is another example where the semantic web perspective is clear (which is fine, as long as it is made clear that the focus is on semantic web type reasoning, which it currently isn't).*

We assume that with "semantic web type reasoning" you refer to DL-based reasoning or in general to any kind of reasoning where ontological axioms are involved. We revised the content of the section to clarify.

> *S3e: There is no "ontology inference process" in the reference architecture. Why is one needed in all stream reasoners?*

The action of S3d should have also addressed this.

> *S3f: How can the separation "safeguard" the complexity? Do you get a different total complexity if you split up the problem in two parts, or are you actually solving a different problem?*

The problem is how to build a solution where CEP rules and ontology axioms co-exist. We can imagine two extreme cases: if we let these rules and axioms interplay, we may end up in situations where the computation goes out of control. On the other extreme, if we strictly separate the CEP-rule and the axioms (first one, then the other), the overall complexity of the rule execution won't increase (it's a sequence of two sub-systems). These two solutions lead to different behaviour and results, but there are several intermediate solutions among them that may be studied. What we are pointing out in this paragraph is the opportunity to investigate this spectrum.

> *S3g: How can a fact still be relevant if it is outside the window? By the semantics of a window only the information inside it matters. If the user had a different intention she should write a different query. The semantics of windows is well-defined, so nothing outside can be relevant (unless the semantics is incorrect).*

We believe that the window is a mechanism for a user to specify a time span related to the computation. In a DSMS scenario, we may take a step further and claim that the direct implication is only the portion of the window captured in the stream that matters. We believe that it may not be true when i) reasoning comes into play and ii) some information happened in the past may influence the current computation (and what can be inferred from the current window content). For this reason, we claim that the role of the window should be revised to take into account such cases. Interestingly, some recent studies in DSMS/CEP are actually going in this direction, such as:

- Olga Poppe, Chuan Lei, Elke A. Rundensteiner, Daniel J. Dougherty: Context-Aware Event Stream Analytics. EDBT 2016: 413-424
- Michael Grossniklaus, David Maier, James Miller, Sharmadha Moorthy, Kristin Tufte: Frames: data-driven windows. DEBS 2016: 13-24
- Alessandro Margara, Daniele Dell'Aglio, Abraham Bernstein: Break the Windows: Explicit State Management for Stream Processing Systems. EDBT 2017: 482-485

> *S3h: How can the semantics of forgetting be more precise? A window is a precise definition of what information is relevant. Yes, there are queries which can not be*

> *expressed with windows, so there is definitely a need for other constructions to capture these. This would extend the formalism which would allow more expressive queries to be asked, but I'm not sure I consider that "improving the quality of the engine answers" since the initial answer was perfectly correct given its semantics.*

Our claim is that it is possible to make further research on windows by a stream reasoning perspective, where in addition to consumption and expiration [24] (new version of the paper), validity comes into play. As we discuss in Section 3.1.3, the semantics of window is related to expiration and is usually associated with consumption. While in a classical DSMS everything works, when reasoning is involved it may not, because it becomes necessary to distinguish between expiration and validity. An expired fact (i.e. that is out of the window) does not imply that it is not valid anymore, and this may influence the current derivations, so consumption mechanisms should take this into account.

> *S3i: In 3.2.1, why is OWL necessary?*

That's right and we agree on the fact that OWL is not necessary. We rephrased the sentence.

> *S3j: In 3.2.2, what do you consider the model to be in stream reasoning? It is not clear to me. If you have a query for selecting some elements from a stream, what is the model?*

We revised this paragraph, and now it should be clear.

> *S3k: In 3.3, yes this has been done. There are some papers on semantic streams such as FUSION'13, IROS'13, SCAI'15...*
> *Daniel de Leng and Fredrik Heintz. Ontology-Based Introspection in Support of Stream Reasoning. Proc. SCAI 2015.*
> *Daniel de Leng and Fredrik Heintz. Towards On-Demand Semantic Event Processing for Stream Reasoning. Proc. FUSION 2014.*
> *Fredrik Heintz. Semantically Grounded Stream Reasoning Integrated with ROS. Proc. IROS 2013.*
> *Fredrik Heintz and Daniel de Leng. Semantic Information Integration with Transformations for Stream Reasoning. FUSION 2013.*

We added the reference to SSL$_T$, and we adjusted the text accordingly, opening the problem of finding agreements, to create standard proposals and to push for adoption.

> *S3l: In 3.4, you state that stream reasoning will not reach the performance of stream processing, what is the different in your opinion? There is no definition of either, so it is hard to argue for this claim.*

Action A6 should have fixed it.

> *S3m: Last sentence in 3.4.1, of course you can do things faster if you have an incorrect solution. Just return nothing for every query, it can't be faster (and probably not much more incorrect either). A reasoner without correctness guarantee is not a reasoner, it is poor hack.*

We rephrased the sentence.

> *S3n: In 3.4, there is plenty of work on how to guarantee the resource usage of stream reasoning in the runtime verification literature. The main concept is trace-length independence. If a reasoner is trace-length independent then its growth is bounded even on infinite traces/streams. A new concept, event-rate independence, has recently been suggested for reasoners that is neither dependent on the trace length nor the rate of events (number of events per time-unit). See for example Online Monitoring of Metric Temporal Logic by Hsi-Ming Ho, Joël Ouaknine and James Worrell from Runtime Verification (RV) 2014.*

We edited Section 3.4.3 and we added a part about online monitoring.

> *S3o: In 3.4, windows should not be used to control resource usage but to define the semantics of a query.*

We agree, there should be a sentence where we have not been clear. Could you please indicate us where it is? We cannot find it in Sec. 3.4.

> *S3p: In 3.5.1, semantic matching as described in the references in S3k is another approach.*

Done.

> *S3q: In 3.5.1, work on sychronization of streams have been done as described in Chapter 7.8 in Fredrik Heintz. DyKnow - A Stream-Based Knowledge Processing Middleware Framework. PhD Thesis 2009.*

Done.

> *S3r: In 3.5.1, how does RSEP-QL relate to the reference model proposed in this paper?*

The reference model proposed in this paper is built on the top of RSEP-QL, and has been initially introduced in Chapter 7 of

- D. Dell'Aglio. On Unified Stream Reasoning. PhD thesis, Politecnico di Milano, 2016.

We added the reference in Section 2.

> *S3s: In 3.5.2, there is plenty of work done in the areas of reasoning about action and change and belief revision that could be used as a basis for further research.*

We added this possibility.

> *S3t: In 3.5.2, there is also plenty of research on reasoning with temporal intervals, Allen's Interval Algebra is probably the best known work, that could be used for further research.*

We do not fully understand this comment. Why do you think that interval temporal reasoning is the solution to cope with noise?

> *The language of the paper can be improved. A detailed list of comments is available as a PDF document.*

Done, and thank you very much for the accurate list of comments.

Reviewer #2

> *As a practitioner in this area, I would have liked some simple use cases that cover the many applications of stream processing, like: traffic analysis of transit information, weather updates or stock market. However, I think that was not the goal of this paper that had a lot of information about the different stream systems reasoning systems.*

Indeed, that was also our feeling. We would have loved to include a running example and use it to compare the various approaches, but that would make the paper exceed by large the space limits. Therefore, we decided opted for a list of requirements hoping that they would serve the same purpose.

Reviewer #3

> *This article appears less to me as a "vision" article, but more as a survey and a critical assessment of the domain's current status. The vision aspect is a rather careful (but well founded) discussion, which I certainly would call useful but not visionary—there are no remotely wild novel ideas in there. Given the broadness of the discussed work, I think it makes sense to reclassify this as a survey paper, which contains a perhaps atypically long "open questions" section, but this it not unexpected given the relatively young age of stream reasoning. Then again, perhaps the importance for the Data Science journal of such labels is perhaps minimal, so maybe it is not that big of an issue. Nonetheless, I would suggest to rethink the title and make it reflect the scope of the article better.*

We are not sure about the importance of labelling the article, so it is probably up to the editor to solve this issue.

> *The connection between reasoning and Big Data analytics, and reasoning and deep learning, is only very lightly touched upon (mostly in 3.2.2). Given the importance of these technologies, it seems necessary to compare more intensively. Especially if this is considered to be a vision for the future, the interaction of these domains seems like a subject too big to ignore.*

We definitely agree with you on this. We mentioned this relation also in other points of the paper, e.g., Sections 3.1.1 and 3.4.2. A comprehensive comparison between Big Data, Deep Learning and Stream Reasoning is out of the scope of this paper, and it requires several studies, given the heterogeneity of solutions you can find.

> *For a survey paper, a major missing area is evaluation/benchmarking. The point is touched upon in the conclusion, but the authors never elaborate on it.*

We got the permission from editors to extend the article in this direction. You can find it in Section 2.4 (any comment on its content will be welcome).

> *The introduction starts with motivating use cases, but it is not clear to what extent these scenarios a) have been solved b) have been solved with stream reasoning c) will be solved in the near future d) will be solved with stream reasoning e) need to be tackled with stream reasoning as opposed to other techniques (especially machine learning, for instance regarding traffic jams).*

In most of the problems we are aware of, Stream Reasoning won't be the only and right solution. On the contrary, we envision it as a set of techniques that can bring improvements, e.g. increase the number of results and simplify the modelling of the problem, as well as the definitions of the queries.

> *What is the source of the 9 R's? volume/velocity/variety are traditional Big Data V's, for instance.*

The requirements come from [37] (new version of the paper). The content is still the same (apart from R9, that is new), but we relabelled them to be easier to understand (e.g., the Big Data Vs) them.

> *What is the relation/difference between R2 and R6? Coping with velocity seems to be directly related to providing answers in a timely fashion, i.e., any system can handle high velocity data if we slow down that data—but the answers will arrive too late. This difference would need to be made explicit; also note that R2 and R6 appear to be coupled in Table 1.*

We agree with you, but while R6 is strictly a property of the processing/output, R2 reflects the ability of an engine to cope with this very peculiar type of input. Being timely fashion (R6) is just the predominant requirement for an engine able to tame velocity (R2). Indeed, it also has to cope with the nature of data streams (i.e., distributed, bursty, etc.).

> *The label for R9, "understand what users want" inaccurately vague. It is not about "understanding" or interpretation of users' intent at all; instead, it is about the flexibility to define certain query types. It is necessary to change this label into something more accurate.*

We replaced "understand" with "capture".

> *I found the comparison paragraph below the requirements rather hard to follow. Table 1 does a better job on the comparison; it would perhaps be clearer to start out with the table and discuss the different types of systems in different paragraphs.*

We revised the introduction, and now it should be clearer.

> *In Table 1: by only indicating the "true" values (V) and hiding the "false" values (F), the table presents the same information in a much more scanable way. Also; try to make the columns narrower so the marks are closer, facilitating visual comparison.*

Done. We also adjusted the caption to help readers in understanding the meaning of the symbols.

> *In Table 1, consider renaming SemWeb into OBDA to align with the caption and the text.*

Done.

> *"When items are represented through RDF (e.g. an RDF graph)"”" => the notion of a “graph” seems strange here. What is the graph? The entire stream, or a single element? In the former case, I would say that the RDF definition of “graph” is not aligned with the notion of a stream; in the latter case, to what extent would it be meaningful to do so?*

We edited the text in Section 2, and now it should be clearer.

> *The two last paragraphs of 2.1.1 are hard to follow; perhaps a comparison table might make sense.*

We decided to omit details and further analyses for the sake of space. As a solution, in the new draft we added two links that interested readers may use to learn more about those languages and systems.

> *3.2.1: Why is it important to investigate other approaches? What is missing fro OWL 2 DL that other languages can/should bring to the table?*

An alternative to OWL 2 DL may bring to different results that may fit specific use cases, or may introduce new time-related operators in the logical language (as in MTL). We rephrased to clarify on this point.

> *3.5.1 push/pull is a discussion that, in my opinion, needs its own place at an earlier point in the article. They are now only casually mentioned, but have a major impact.*

We definitely agree on this, but we decided to not treat this point for the sake of space.

> *The statement that “stream processing has always coped with noise” seemingly contradicts Table 1.*

We believe that this misunderstanding rises from the absence of a stream processing definition. Action A6 should have addressed this point.

> *This section, unfortunately, disappoints, especially for a vision paper. What does the analysis of the article mean to the audience? The third paragraph summarizes a couple of trends and predictions, but does not really point to concrete action points or emerging fields that require a close follow-up. I expect more advice and inspiration, and perhaps even criticism or suggestions on a meta-level. For instance, the authors write that “it is necessary to develop benchmarking and evaluation activities” (a topic, by the way, that is not covered elsewhere), without pointing at a direction in which authors should think/look to do so.*

The third paragraph is a summary of the content of Section 3, which is an overview of the points we believe to be relevant and that are still open for research. We introduced an explicit link to Section 3, which should help reading. Regarding the benchmark part, the editor granted as the extra space required to add it (see Section 2.4).

> *Similarly, the last paragraph about "real problems and scenarios" is very limited. Where will stream reasoning make a real difference according to the authors? This is a very important question with a visionary aspect that is absent from the conclusion.*

We believe that the goal set in [37] (new version of the paper) is still the open goal that stream reasoning should reach. The contribution should be in improving real-time analytics on streaming data by exploiting the semantics. We moved the sentence at the end, hoping that it can make the job.