LEARNOVATE
Leading Learning Innovation

xAPI How to Guide

Authors:
Mirjam Neelen – Learning Lead
Dr. Benjamin Heitmann – Technical Lead

TECHNOLOGY CENTRE
ENTERPRISE IRELAND
IDA IRELAND SUPPORTED

Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

1 Contents

1. Introduction

Based on previous analysis, you have identified that xAPI is the right solution for your organisation. This document aims to support you in identifying how to best leverage xAPI. There are six main categories that you need to consider:

1. Be clear on **why** you want to use xAPI.
2. **Learning Designers** need to learn to **design for statements**.
3. **Learning Design, Data Analytics and Software Engineering experts** need to collaborate closely.
4. Realise that there will be a **steep learning curve**.
5. Recognise the **need for growth and change management**.
6. **Align privacy/security provisions of xAPI** with legacy and future requirements.

In the following, we will describe each recommendation in more detail. In particular, we will provide a detailed motivation for each recommendation. We will also describe how we applied the recommendation to our own proof-of-concept prototype at a high level.

If you would like to discuss these recommendations in more detail, or if you have questions about how these recommendations would apply in detail to your particular use case and your organisation, please make sure to get in contact with the Learnovate Centre.

2. *How to Leverage xAPI: Principles to Keep in Mind*

2.1. *Be clear on why you want to use xAPI.*

There are many reasons why you would want or need to use xAPI. Many of those reasons are outlined in Learnovate's xAPI Checklist. The main areas to think of are:

1. Learning design
2. Learning analytics and data science
3. Software architecture, design and implementation

They are all equally important. The volume of data being written to the LRS has the capacity, even in a small ecosystem, to be large, especially when data aggregates over time. As a result in larger implementations of xAPI, the resulting large volumes of data means that it becomes harder for those managing learning to recognise the meaning behind the data and draw conclusions. Therefore, if your goal is to assess or evaluate learning impact, it is important to design for what you are trying to accomplish beforehand to ensure that you know what data you want to query and why.

1.1.1 *Learnovate Example*

Learnovate started with the learning design. We wanted to design for a specific learning need, in this case in a K12 scenario. We had an existing demonstrator app, called SkillTrack!. SkillTrack! aims to support the practice, development and self-review of 21st century, or transversal skills in order to make the implicit more explicit. This demonstrator app focusses on students. However, Teachers also need support to make the implicit explicit without compromising valued content instruction time. In addition, Teachers need to be able to develop an understanding of student experience around 21st century skills and how certain instructional choices can help promote and develop them. A tool for Teachers to help them reflect upon their instructional choices and teaching practice could be useful. Teachers can then strengthen the classroom experiences available to their students to help develop literacy, development and practice.

Challenge for Teacher: How to become more aware of any gaps between student and Teacher perception of when and where the 21st century skills are being experienced within the instructional activities in the classroom. This awareness is useful for the Teacher so that he/she is better able to strengthen the development of students' 21st century skills through the opportunities provided within the instructional activities.

Figure 1 shows, in short, this is how it works:

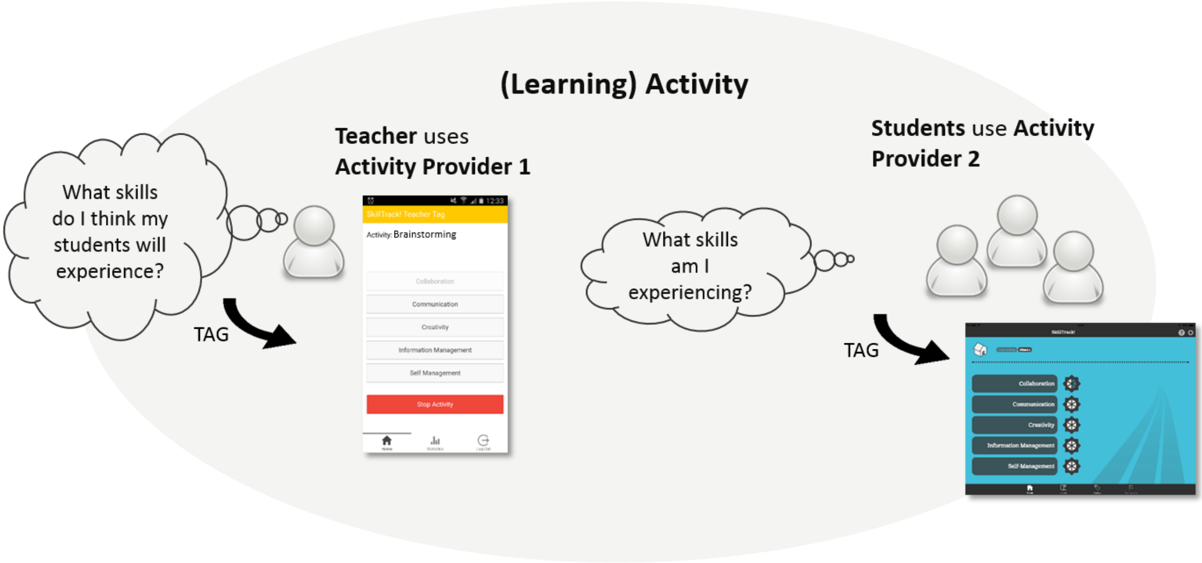

Figure 1. Learnovate's Use Case Scenario Example

- The Teacher sets up an instructional activity in the app (e.g. brainstorming activity) for a specific group (e.g. English Year 2).
- During the instructional activity, both the Teacher and the students tag skills. The Teacher tags the skills they think their students will experience during the instructional activity, the students tag the skills they experience during the instructional activity. For a brainstorming activity, this could be creativity, collaboration, communication.
- After class, the Teacher can compare the skills they have tagged with the skills that the students have tagged. In other words, to what extent do students and Teacher tag the same or different skills for a particular activity. If the tagging is very different, the Teacher needs to consider the possible reasons and take action.

2.2. Learning Designers need to learn to design for xAPI statements.

After identifying the learning need, in other words, why learners would need to complete the learning activities, for example complete a course or contribute to a social learning platform activity, the next step is to identify the learning analytics that would give you the insights that you are looking for.

After you have identified the learning analytics that you are looking for, you can work backwards and design the xAPI statements. After all, you can only design the correct statements if you know what you are trying to track and why.

Learning experiences are stored as statements. Figure 2 shows the three basic, but fundamental elements of statements; that is actors, verbs and objects. The context field is optional and provides a place to add contextual information to a statement.
An *actor* can be an individual or a group with a unique ID. For example:

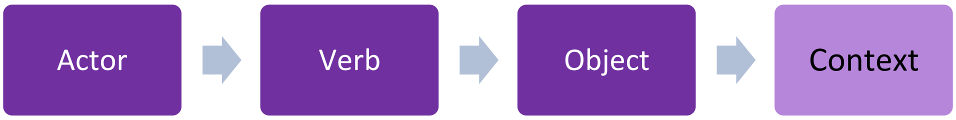

Figure 2. Elements of xAPI Statement

- Actor: Ben
- Unique ID: ben@tcd.ie

A *verb* describes the action performed during the learning experience. For example:

- Answered
- Attempted
- Completed
- Shared
- Evaluated

The *object* usually describes the actor’s learning experience. For example:

- Business Seminar
- Chemistry 101 Lesson
- Virtual Simulation

Richer statements are potentially more meaningful and easier to interpret. Richer statements include context, such as

- Ben shared a document on 'Getting Better at Customer Service' on Discussion Forum 'Customer Service' via iPhone.

2.2.1 Learnovate Example

In Learnovate's case study, one of the main learning objectives was for the Teacher to identify **any gaps between student and teacher perception of when and where the 21st century skills are being experienced within the instructional activities in the classroom**. This means that we needed to be able to *compare* the skills that the students tagged with the skills that the Teacher tagged.

Table 1 below summarises the statements that we used in our use case.

Action	Actor	Verb	Object	Context
Teacher starts an activity.	Teacher	Started	Activity "Brainstorming Session"	English Year 2
Teacher tags a skill for an instructional activity.	Teacher	Tagged	Skill "Communication"	Brainstorming Activity English Year 2
Student tags a skill.	Student	Tagged	Skill "Creativity"	English Year 2
Teacher stops an activity.	Teacher	Stopped	Activity "Brainstorming Session"	English Year 2

Table 1. Learnovate's Use Case Statement Examples

2.3 Software Engineers, Data Analytics & Learning Design need close collaboration.

Adopting xAPI in an organisation requires the availability of expertise in the areas of (1) learning design, (2) learning analytics and data science and (3) software architecture, design and implementation. All of three areas of expertise are required in order to leverage all potential benefits of an ecosystem based on xAPI. HR would need to be involved as well to, for example, ensure that xAPI implementations are aligned with business objectives.

In addition, for adopting xAPI, close collaboration between the team members representing these areas of expertise is required, as no single perspective on how to adapt and implement xAPI in an organisation, will be able to unlock the full potential of xAPI.

First, an organisation should have expertise in the area of **learning design**. This is required to identify learning needs. These will inform which learning events should be aggregated across all systems in order to enable meaningful insights into the aggregated data. In addition, expertise in the area of **data science** is required for statistical analysis and machine learning approaches to be used to provide further analytics results and to complement the learning design expertise. Finally, expertise in **software engineering** is required to design and implement the infrastructure of xAPI (such as the LRS), to implement the modifications to existing systems, and to add new components such as a middle ware, should this be required.

2.3.1 Learnovate Example

To implement the Learnovate case study, experts from all three mentioned areas were made available from our team. In addition, the team collaborated closely, as decisions in one area had to be informed by the requirements of all other areas.

Learning design: The Learning Designers provided the motivating use case, as described earlier in this document. In addition, the list of requirements for all user-facing components was provided by the learning designers, e.g. they specified which questions the analytics should answer and how to visualise the data. For instance, the Learning Designers made it a requirement to allow a Teacher to compare the skills tagged by the Teacher and the skills tagged by the students so that these tags could be compared and any gaps could be identified.

Learning analytics and data science: Expertise in the areas of learning analytics and data science was required in order to determine the requirements for aggregating and querying the xAPI data. In particular, our requirements in terms of learning analytics informed our choice of Learning Record Store (LRS), as we had two candidates available which differed in their query API. We selected the LRS with the more scalable query API, as this was required in order to fulfil the learning analytics requirements.

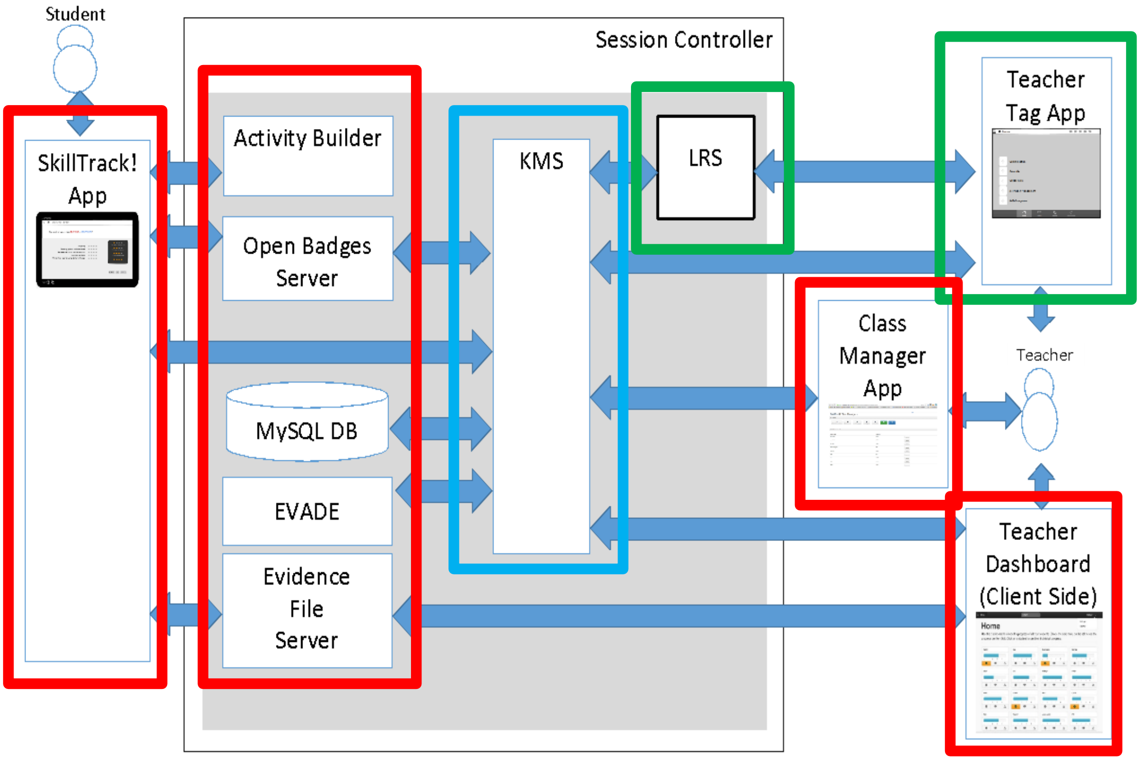

Figure 3. Technical architecture of the proof-of-concept prototype used in the Learnovate case study. New components are green, modified components are blue, components which cannot be modified are red.

Software architecture, design and implementation: Finally, expertise in relation to software architecture was required to select an approach for integrating the existing legacy components into the xAPI ecosystem. In particular, as all legacy components were interacting with a dedicated middleware already, we extended this existing middleware to translate the relevant data into xAPI statements, and send these to the LRS.

Figure 3 shows the technical architecture of the proof-of-concept prototype used in the Learnovate case study. In the diagram, new components are green, modified components are blue, components which cannot be modified are red. The reason why most components could not be modified, was that changing these components would have required resources outside of the scope and time constraints of the Learnovate case study.

In order to minimise the impact of adding xAPI to the existing legacy components, one component was modified. This is the Knowledge Management System (KMS), which was used in the legacy system as the central middleware, and all other legacy components were already communicating with the KMS.

We added two new components, which are the Learning Record Store (LRS), and the new App Prototype (called Teacher Tag app in Figure X above) for mobile phones. xAPI statements are stored in the LRS via the Teacher Tag app, and via the KMS, which translates all relevant data to xAPI statements.

2.4 Realise that there will be a steep learning curve.

Introducing xAPI into an organisation can seem like a minor task, or a simple upgrade. However, it would be a mistake to underestimate the effort and planning to adapt xAPI in an organisation due to several important factors:

- xAPI is not “just” one standard. Instead **xAPI is a collection of standards**, which describes how to use several other standards together. These standards are REST APIs, JSON for data transport, OAuth or Basic HTTP Authentication for security, as well as the unique data model of xAPI statements. **Adopting xAPI in an organisation also means adopting all its constituent standards as well.**
- Together, xAPI and its constituent standards describe a blueprint for how to build not one system, but for a system of systems. Such a system of **systems is usually called an “ecosystem” in software architecture**. The difference lies in the fact that xAPI describes not only how to store data, but also how different systems should transmit data between each other. In addition, **xAPI prescribes a specific architecture for ecosystems using xAPI**, in which the LRS forms the centre of the ecosystem, as it must store the xAPI statements generated by all other systems in the ecosystem. Therefore, adapting xAPI in an organisation can touch many of the (legacy) systems in an existing ecosystem if they have to be changed.
- Adapting xAPI requires expertise in at least **three separate areas of expertise**, which are learning design, learning analytics and data analytics, as well as software

architecture, design and implementation, as described earlier in this document. In addition, it is important to note that **implementing xAPI in your organisation might require upskilling of all 3 types of experts**. For example, Learning Designers need to learn how to design for xAPI statements, Data Scientists need to learn how to evaluate learning impact, and Software Engineers need to understand how the learning needs impact the technical architecture and functionality.

- **Modelling of xAPI statements also requires a different way of thinking about the data** collected in an xAPI ecosystem, as discussed earlier in this document. The main complication is that xAPI statements are about learning activities only. This might be different in subtle details from existing ways to capture data about learning in an organisation.
- Current off-the-shelf implementations of **Learning Record Stores (LRS) are at an immature state in their software life cycle**. As such, the effort of selecting and maintaining an LRS in an organisation should also not be underestimated. First, careful analysis of the requirements of an organisation in regards of an LRS are required, in order to select the most suitable available LRS. Then the next step is to determine if the LRS needs to be modified to implement any missing requirements. Lastly, the strategy for integrating the LRS into the existing ecosystem needs to be considered. Existing components / systems can be changed to communicate with the LRS, or a middle ware can be introduced to translate between existing systems and the LRS, or a mixture of both strategies can be employed. All in all, due to the immaturity of existing LRS, this step will add to the overall effort of adopting xAPI.

Taken together, all of these factors contribute to a steep learning curve for adopting and implementing xAPI in an organisation. This should be considered when allocating time and resources for the transition to xAPI.

2.4.1 Learnovate Example

There are multiple ways in which the steep learning curve of xAPI was visible during the Learnovate case study, even though our project was a relatively simple one.

For starters, **setting up the LRS** was a considerable amount of work for our software engineers. Two candidates for the LRS were available, and setting up a test installation took between two and three man-days for each one. One of the LRS had almost no documentation on the setup, so instructions had to be collected from unofficial weblogs. Then, after collecting the technical requirements for selecting one of the two LRS, we had to consult an unofficial channel again in order to compare the query APIs of the two LRS. The LRS which we ended up using had a high performance query API which extended the query API as mandated in the xAPI specification, however the documentation for this extended query API was unwritten, so testing it took another couple of person-days.

From a technical perspective, the next step was to integrate the LRS into the existing ecosystem. We re-used all the existing backend components of the SkillTrack! app for self-assessment of 21st century skills. However, after reviewing the security options of xAPI, it became apparent that this would be another crucial requirement to inform the choice of the LRS. xAPI provides a choice of two different security protocols to secure the communication between components: OAuth and Basic HTTP authentication. We quickly decided that implementing OAuth brought too much implementation overhead, as all existing components would need to be changed to support OAuth. However, only one of the two available LRS supported Basic HTTP Authentication.

Taken together, the choice of query API and the choice of authentication mechanism were the most important requirements for making the decision for an LRS. Luckily for us, the same LRS supported both the extended query API and Basic HTTP authentication, so we deployed this LRS for the Learnovate use case.

After selecting the LRS, we had to decide on a strategy to integrate the existing components with the LRS. This is described earlier in this document in Section 2.3.1.

Lastly, the task of designing for statements added to the learning curve. As described in Section 2.2.1, only four statements are required for the use case in the Learnovate case study. However, we discovered that the process for designing statements involves three steps:

1. **Identify learning activities** to track and their context. In this step, the use case is analysed to determine which learning events are happening in the use case, and to identify their context.
2. **Map** learning activities **to statements**. In this step, all events are collected and grouped, after which verbs are assigned as “lowest common denominator” of event groups, or to individual events if an event falls outside of any group.
3. **Standardise** between all activity providers. In this step, the JSON syntax for all xAPI statements from the previous step is defined. Then these xAPI statements are implemented in all participating data providers (call activity providers in the xAPI standard).

While following this process can simplify the design of the xAPI statements, the process itself is still not very intuitive and can lead to debates around the right choice of statements and their syntax.

2.5 Recognise the need for growth and change management.

If your organisation is expecting to maintain an ecosystem with many data providers and fast growth, then xAPI is a good candidate.

To integrate data from internal and external sources into an ecosystem based on xAPI, the data from these sources must be converted to xAPI statements. These xAPI statements should use the vocabulary and the syntax as defined by the organisation maintaining the xAPI ecosystem and the Learning Record Store (LRS) at the centre of the ecosystem. However, it can be challenging to formulate a vocabulary that fits not only the requirements and specifications of current data providers, but also those of future data providers.

For instance, if you designed the vocabulary for your xAPI ecosystem to only contain verbs related to financial training courses, it might be hard to also express learning events related to health and safety using this vocabulary. In this case, you will have to add new verbs to your vocabulary. Repeating this process for many different data providers over a long time might result in problems related to aggregating and interpreting this data. Therefore, the vocabulary and the xAPI ecosystem of your organisation will have to be re-evaluated from time to time to determine strategies to manage growth and change in your ecosystem.

One of the main goals of xAPI is to enable and facilitate ecosystems with many data providers and fast growth. However, you will have to manage the growth and change of your ecosystem yourself, as this goes beyond the scope of the xAPI standard as such. For instance, after adding data providers from many different domains (first finance, then health and safety, then legislative procedures) you might decide that a different set of verbs is better suited as a lowest common denominator for your ecosystem. In this case you would need to finalise a new version of the vocabulary, send it to all data providers, and enforce the use of this vocabulary by all systems in the ecosystem after a specified point in time.

2.5.1 Learnovate Example

This recommendation does not really apply to the Learnovate case study, as we have a limited number of two activity providers adding data to our LRS. In addition, we have been operating our LRS only for the three months since the inception of our case study. So we did not really have to look into growth and change management for our own ecosystem.

However, one of the main points when marketing xAPI is the idea of “life-long learning”. xAPI aims to support the goal of life-long learning by aggregating learning events and experiences of an individual over a very long time span, ideal over his/her whole life.

While this goal is within reach of the technical capabilities of xAPI, it is only practically achievable if the ecosystem in which the learning data is collected is carefully maintained. In addition, growth and change management as described in this section are an absolute requirement for any ecosystem in which life-long learning is encouraged. Without making growth and change management a first-class priority it will be almost impossible to relate and compare learning events collected at different points in time.

Imagine a learning event related to successfully completing a first aid course in 2015, and a similar event for successfully completing a first aid course in 2025. Even if both statements use the same verb and the same syntax, how will an analytics component be able to deduce that both events are about first aid, without some sort of subject categorisation system? This subject identification system would need to guarantee that the same subject identifier is used for the concept of “first aid” in 2015 and 2025.

To further complicate the matter, if such a subject identifier is agreed upon, then it might not be used if the first aid course in 2015 was organised by a different first aid training organisation as the one in 2025. There might be no incentive for an organisation to check for existing and reusable subject identifiers for use in xAPI statements generated by that organisation. In such a case, no comparison of learning events would be possible without having a human expert add the connection between the two first aid courses to the LRS.

2.6 Align privacy/security provisions of xAPI with legacy and future requirements.

xAPI provides a blue-print for federated ecosystems in which data about learning events is shared with a Learning Record Store. One part of this blue-print prescribes the use of two standards to address security and privacy needs in an xAPI based ecosystem. These two standards are OAuth and “Basic HTTP Authentication”. Every organisation which implements an xAPI ecosystem is advised by the xAPI standard to implement one of these two security standards to secure the learning data.

1. “Basic HTTP Authentication” is based on the same standards which are used in the wider IT area since the inception of the World Wide Web in the 90s and presents a baseline in terms of security. It allows controlling the access to data; however it does not allow for more advance capabilities such as authorisation of 3rd parties to access data on behalf of a user, or the revocation of authorisation for 3rd parties. Using “Basic HTTP Authentication” is suitable for ecosystems without external data providers.
2. In contrast, OAuth goes beyond classical authentication mechanisms of the World Wide Web. It allows a user to authorise external 3rd party providers to access his data until a point in time when the user decides to revoke this authorisation.

In summary, xAPI provides an integrated approach to secure the data of an xAPI based ecosystem. However, this will also require adding support for the relevant standards to all components of the xAPI ecosystem.

2.6.1 Learnovate Example

The choice of security standard was an important consideration, for the decision of how to integrate the xAPI LRS into the architecture of the Learnovate case study.

Choosing OAuth would have enabled more complex use cases for authorising external 3rd party data providers in the future. However, choosing OAuth would have also required changing most existing legacy components. The estimated effort in terms of software engineering resources was quickly deemed as unrealistic for our proof-of-concept prototype.

Therefore, we selected Basic HTTP Authentication for the proof-of-concept prototype, as it was already supported by the existing legacy components.