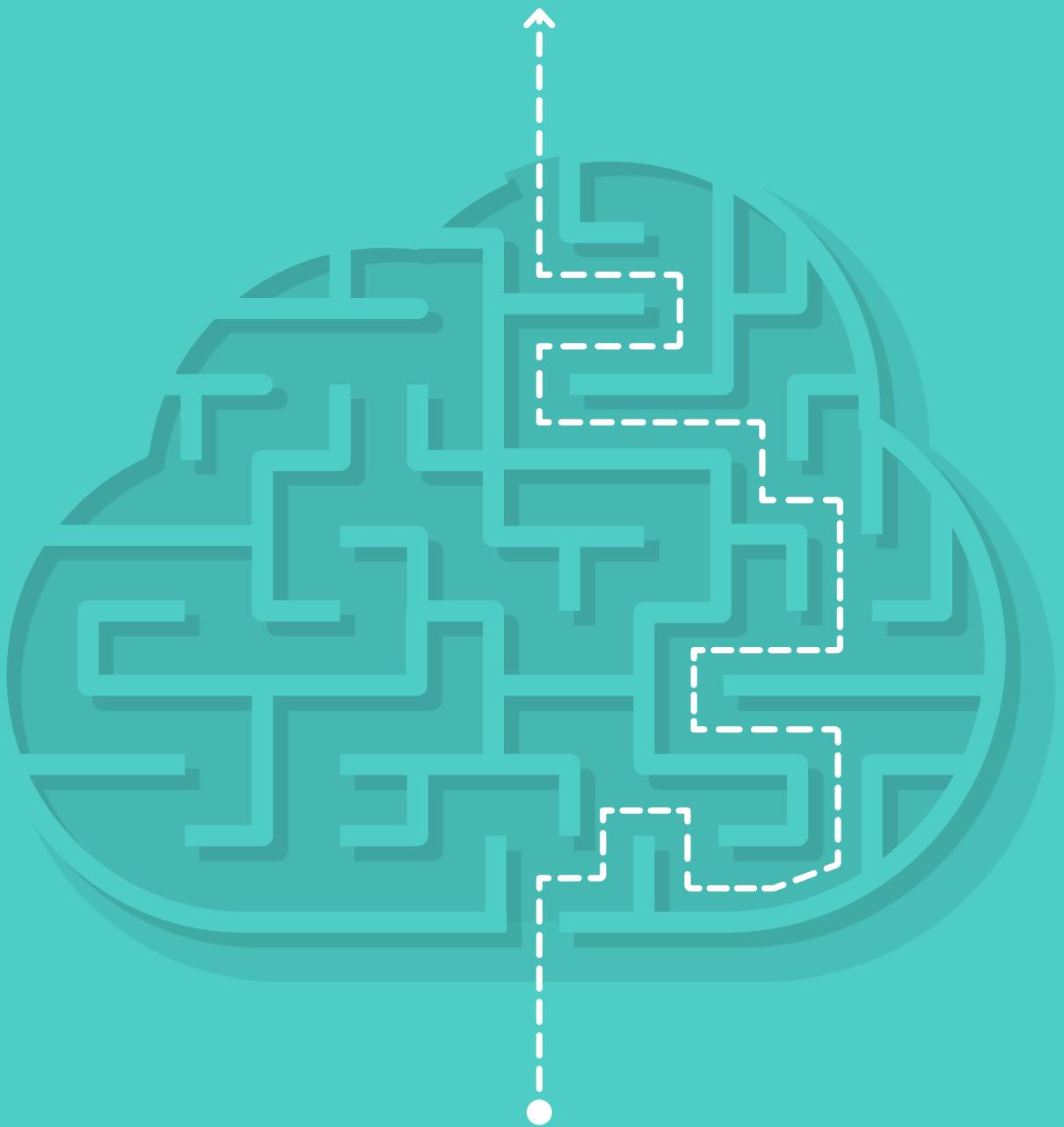


THE LEARNING TECHNOLOGY MANAGER'S GUIDE TO xAPI



Getting Started ▪ Case Studies ▪ Professional Insight



Learning Locker
HT2Labs

LEARNING RECORD STORE

The Learning Locker dashboard is a powerful tool for managing and analyzing learning data. It includes a variety of widgets such as:

- Mystery shopper score by country (Bar Chart)
- Onboarding Journey (Heatmap)
- Leaderboard (Horizontal Bar Chart)
- When do the big hitters shine? (Bar Chart)
- Activity over time (Scatter Plot)

learning
technologies
awards 2017

THE MOST INNOVATIVE NEW
LEARNING TECHNOLOGIES PRODUCT

Learning Locker is the world's most installed Learning Record Store. Available open source or with our Enterprise version, experts use Learning Locker to aggregate, automate, and analyse learning data.

Now it's your turn.

CONTENTS

2 INTRODUCTION

Ben Betts, CEO of HT2 Labs, introduces the guide with a short piece on the role of data in learning technologies.

4 INTRODUCING THE EXPERIENCE API

In part 1 we get acquainted with the Experience API (xAPI), where it came from and how it relates to existing technologies. We'll explore how we can get systems sending data in the xAPI format and how organisations use xAPI data to improve their businesses.

15 INTRODUCING THE LRS

Part 2 of our guide will introduce the concept of the Learning Record Store, or LRS. We'll investigate key questions like how your LRS might relate to an LMS, where you might store your data and how you'll secure it. We'll also introduce our LRS, Learning Locker, its key features and pricing.

22 DEVELOPING xAPI STATEMENTS

We'll get a bit more technical in part 3 as we look at the anatomy of an xAPI Statement in more detail. We'll go over each of the key elements that make a Statement and give examples. Finally we'll give you three tips to make interacting with an LRS just a little bit easier.



35 CASE STUDIES

Our final section is dedicated to case studies of early adopters who are using the xAPI today. You'll read about a range of corporate and education uses, including relating training activity to sales performance, increasing learner success and creating a gamification system.

CREDITS

Authored by

Ben Betts & Ryan Smith

Edited by

Andie Coupland

Layout & Design

Ollie Newton

Produced by

HT2 Labs
www.ht2labs.com

COPYRIGHT

This work is licensed under version 4.0 of the Creative Commons with the restrictions CC-BY-SA. You can use, study, apply, redistribute and create derivative works based on this document as long as you give proper attribution to the author, retain the license notice and release any derivative works under an equally permissible license. See creativecommons.org for license wording and more info.





IF ONLY EVERYTHING WORKED TOGETHER

2

In an age of where my phone knows when I'm leaving the office, checks the traffic home and tells me how long it will take, it doesn't seem like much to ask that our learning systems work together. They don't seem to be the most complex pieces of software in existence, so why is it that sometimes they behave like it?

There are a range of problems afflicting the learning systems we've gathered over the years. From organisations with multiple LMS's, to those also deploying talent management systems, knowledge bases, social networks, social learning portals, mobile apps and simulations. The list of potential software that the L&D department could use is growing by the day. But, for many people, it's a real problem just trying to get the existing systems talking to each other. Never mind introducing something new and potentially impactful; the old stuff just doesn't really work.

For learning technology managers tasked with realising the L&D department's ambitions, the pain is real. Whilst leadership wants to get behind 70/20/10 principles and start implementing next generation learning technologies, the mere thought of migrating to a new LMS is enough to make most learning technology managers want to cry.

But with an increasing marketplace for 3rd party tools that the L&D department wants to integrate with the business, learning technology managers are learning that they need to adopt a more service orientated architecture, plugging lots of small, best-of-breed systems together, to offer up a playbook of tools and technologies that HR colleagues can use to improve learning outcomes.

Establishing a service orientated architecture often starts with establishing a method of getting data from one system to another; data reconciliation is a big, expensive business. With no single source of 'truth' in the organisation, we find ourselves trapped.

When systems cannot share data it becomes impossible for them to work together. The LMS doesn't know when the simulation is complete. The App can't track back to the social learning platform, and the social learning platform doesn't do SCORM so we can't track that either. Content from 3rd parties? Forget about it...

Getting control of learning data should be the top of the 'to do' list for every L&D department.

Without the control that a single, accessible source of record can give, you have no base from which to start making things work. Time is of the essence. You need a basis in historical data to start benchmarking current performance against past results. Every day you are not standardising the collection of learning data is another day you are letting the future lifeblood of the L&D department slip away.

This guide will introduce you to the concept of the Experience API, or xAPI as it is more commonly referred to; a specification for collecting, storing and querying learning data that can help you establish a new learning technology architecture for your business. It's aimed at learning technology managers working within L&D departments (but is equally relevant to those working in other areas). We'll take it easy to begin with and slowly ramp up the complexity to give you the complete guide to getting started with xAPI.

What are the top three data issues faced by Learning Technology Managers?

1

Incomplete data. It feels like you must be tracking everything but somehow key data points get lost along the way. You find yourselves with all sorts of expensive compliance issues.

2

Incorrect data. Potentially even worse, the data you have might not be timely or even accurate. With no real standard way of collecting data in the organisation, it's up to your vendors to have done a good job in standardising and storing data.

3

Siloed data. You've got no way to get data from one system to another, without calling up IT. Adding in another system just gives rise to more headaches with the first two problems. And migrating to a new LMS? Forget about it!

DR BEN BETTS - CEO, HT2 LABS

Ben leads the passionate team at HT2 Labs and is a globally-recognised thought-leader in Learning Technology with more than 15 years industry experience.



4

INTRODUCING THE EXPERIENCE API

The Experience API or xAPI for short; is a specification document created by a consortium of learning experts led by the Advanced Distributed Learning Initiative (ADL), which is a US Government Programme. Whilst the document was being formed its prototype name was 'project Tin Can', but these days it is known by its official name, xAPI.

xAPI is a document that, if you follow it, will give you a method for how you can communicate learning activity data between one system and another. It is a derivative of the term 'API', or Application Programming Interface. Before the advent of API's, getting two systems to talk with each other required a large amount of code development and integration time. API's have transformed this relationship, meaning that basic integrations between different software packages are now relatively trivial.

You'll use APIs everyday, whether or not you notice them. At its most basic you can consider an API a bit like a Question and Answer stream: One system will ask the other system a question, and get a response in a format it can interpret. So, one system might ask another, "*can you show me a list of all the registered users on your system?*" and the other system would respond with a formatted list of registered users. By using this Q&A style format we can limit the access one system has to another - your system will only answer the questions for which it is authorised to give out the answers, you don't have to give complete access to a database.

One problem with APIs is that they can be quite specific; I can come up with the way I want my system to answer your questions and you'll have to customise your questions to match the information I'm willing to give. Simply having an API isn't enough; it must be really well documented or even better, subscribe to use some wider standard that we can all learn about, saving ourselves the hassle of using a new API each time we want to make a connection.

The xAPI is an example of the latter; a group of people coming together to decide on the language we will collectively use to build our interfaces - a standard operating procedure. Because xAPI has been built by learning people, it uses language that allows us to talk about things we do in learning - like querying for assessment results or understanding who was a learner's tutor.



*This is why
we use the
Experience API
(xAPI) over and
above any other
API: it's really
relevant to what
we do.*

THE RELATIONSHIP BETWEEN SCORM AND xAPI

We do already have systems in place for tracking learning, the most common being SCORM (the Shareable Content Object Reference Model) - which allows you to package up content to deploy on a Learning Management System (LMS) and then track who uses it.

There are similarities between SCORM and xAPI. The same government department created both. In places they overlap with what they do and how they do it. You could even say xAPI is an evolution of the SCORM principles, but they are mostly quite different. You wouldn't necessarily replace SCORM with xAPI and vice versa.

SCORM allows you to specify a package of content and the sequence in which it must be navigated, and other things too. With xAPI you cannot do this.

The API setup around SCORM is based on the principle that you have a package of learning content and you want to load it onto an LMS for users to launch and be tracked using.

But learning has evolved and we're no longer all about logging in to the LMS to launch learning content. Now we learn from micro-content, located in a wide variety of places. We use mobile phones, simulations, games and more.

We're also increasingly cognisant of the impact of 'informal' learning experiences; stuff that happens outside of the sphere of L&Ds control. How can we account for this?

SCORM has become inadequate for our current and future needs. Learning doesn't always happen on the LMS. Enter xAPI.

xAPI gives us a looser framework that allows us to communicate between any two systems, not just between learning content and the LMS.

In creating the xAPI, its authors couldn't readily put any IT system out of scope - we could be just as easily learning on a phone, or a TV, or in real-life. This is where xAPI thrives, allowing us to understand what occurred on one system, in another system, in a human-readable way.

In the short-term we still have a need for SCORM. Right now it isn't easy to package content up using some of our most popular authoring tools and get it working with your LMS. The xAPI community is working on this through initiatives such as 'CMI-5'.

But it is to the future, where we no longer consider 'packaging' learning content up, that most of our attention is focussed.



Expert View

Specification Writer,
Aaron Silvers

SCORM began as a specification in 1997. It's 20 years old, and still SCORM Version 1.2 is dominant as the means by which eLearning exists -- and when we think about working with learning technology systems, it is still all about courseware and SCORM.

SCORM Version 1.2 is so popular and so adopted that people for the most part never even bother with SCORM 2004. This is regardless of the resources and years that went into improving SCORM from 1.2 to 2004.

Even by 2009, SCORM Version 1.2 was so entrenched that no vendor really wanted to touch SCORM at all. If anyone was going to address mobile, social media, games, simulations or any other use cases, it would need to be outside of SCORM.



HOW DOES THE xAPI WORK?

In an xAPI world we talk about ‘Activity Providers’ (APs), Learning Record Stores (LRSs) and Activity Consumers (ACs).

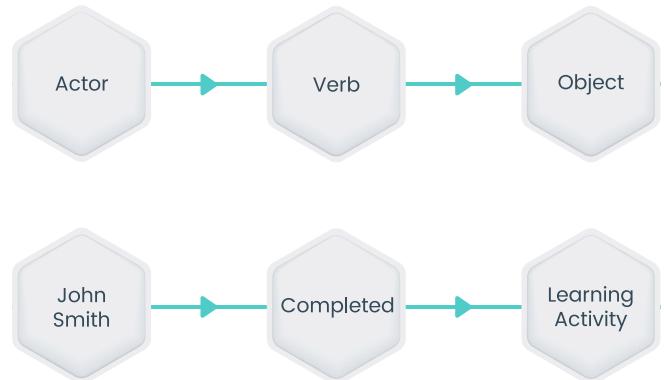
xAPI format and submit it to the LRS. APs are systems and applications on which learning activities and events take place. Learning content, learning portals, apps and more can all behave as ‘Activity Providers’. In an xAPI ecosystem we expect many Activity Providers to be submitting data to the LRS at the same time.

Learning Record Stores are databases that verify that the input matches the xAPI specification, storing all valid data for retrieval by Activity Consumers, or by administrative users who wish to access the ‘raw’ xAPI data for analysis.

Activity Consumers are similar systems to Activity Providers (an AP could in fact be an AC as well), in that they are typically systems and applications that modify the user experience based on xAPI data. This might be an LMS that ‘checks off’ a completed learning activity, because that activity appears in the LRS. Or it could be something more complex; a leaderboard, a badge issuing system or learning content.

For example, if you had a piece of eLearning content that was set to track via xAPI we would consider that the ‘Activity Provider’; it’s the source of the learning activity data. This activity data would then be sent to the LRS in the form of an ‘xAPI statement’. The LRS would accept the data (providing it is valid) and store it. Later on, an administrator may visit the LRS to check on a learner’s progress, which is stored here. This data might also be consumed by another piece of eLearning content, making it both an Activity Provider and Consumer. When acting as a Consumer, the eLearning content could advance the user onto a point deeper into the learning activity, bypassing content that the learner probably already knows, if the relevant previous experience is found in the LRS.

eLearning is just one example of an activity provider. Learning Management Systems (LMSs), mobile apps, portals, smart devices and other software packages, like a CRM, could all be sources of xAPI data. Almost all of these sources could also be consumers under the right circumstances. We also see analytics programs, certification and badging platforms and other Learning Record Stores (LRSs) as likely consumers of xAPI data.



WHAT IS THE xAPI FORMAT?

If you’ve read much about the xAPI previously, chances are you’ve come across the idea of an ‘xAPI Statement’, that is, the manner in which we transfer data between the Provider, the LRS and the Consumer. That statement takes the form of what we call a ‘triple’: An Actor (John Smith), a Verb (completed) and an Object (a Learning Activity).

To extend the meaning of the statement, we add context about John Smith, about the Verb, the Object and about the wider circumstances in which the activity took place. This context might include details like where John Smith was in the world when the activity occurred, what course the learning activity was part of, and how long John spent on the activity, for example.

Using this simple format, you can understand what’s been done, anywhere - be it on an LMS, application, mobile device, even on a point of sale device. In fact pretty much any device that can be connected to the internet can be translated into this format.

Technically speaking, xAPI statements are parsed using JSON - JavaScript Object Notation. JSON looks a bit like XML but it is generally simpler to follow and more lightweight - it takes fewer characters to write what you mean. Whilst it might look a little odd at first, JSON is quite readable as a format, as you can see on the next page.

Take a look at the text on the right. That's how we say 'John Smith completed Activity 1' using JSON. Notice that even at this stage we've got quite a bit more detail than the headline 'triple':

- We know that John Smith is a person that can be identified as by his unique ID '123' at '<http://www.example.com/users/>', as well as the name John Smith (which is kinda helpful as we might imagine a few John Smith's kicking about).
- We know that when we say 'completed' we actually mean the definition of completed found at <http://adlnet.gov/expapi/verbs/completed>
- And we know that John Smith worked on a learning activity that can be found at <http://www.example.com/activities/1> - after all, there could be millions of learning activities that define themselves broadly as number '1', but there can only be one learning activity at that web address at a given time.

```
{  
  "actor": {  
    "objectType": "Agent",  
    "name": "John Smith",  
    "account": {  
      "name": "123",  
      "homePage": "http://www.example.com/users/"  
    }  
  },  
  "verb": {  
    "id": "http://adlnet.gov/expapi/verbs/completed",  
    "display": {  
      "en-GB": "completed"  
    }  
  },  
  "object": {  
    "objectType": "Activity",  
    "id": "http://www.example.com/activities/1"  
  }  
}
```

To avoid issues of data being tampered with, or otherwise altered after it has been sent, the Statement API is immutable - that is, you can't edit or delete a statement after it has been sent. If you've genuinely inserted a statement by mistake (or with the wrong detail) then you can use the 'Void' technique to render the previously stored statement redundant (more about Voiding later). But you won't find an LRS with the ability to directly edit or even delete a statement as it's not a feature of xAPI.

There's a lot more to statements than that, and we'll go into more detail about additional context later on, but statements aren't the only thing we can talk about using xAPI. In fact the xAPI specification actually gives us four different API 'endpoints' to which we can submit data and get data back. These are the Statement, State, Agent Profile and Activity Profile APIs.

The **Statement API** allows you to create statements and retrieve one or many statements that you previously created, as seen above.

The **State API** component of the xAPI allows you to associate the relationship between a user and an activity in a given 'state' such as a user progressing through a course, e.g. User X is on level 4 in Tetris. The xAPI specification states that "*generally, this is a scratch area for Activity Providers that do not have their own internal storage, or need to persist state across devices*".

The **Agent Profile** is the component of the xAPI that allows you to store some additional data against a profile, such as whether that profile is part of a specific group that has access to additional settings or information.

The **Activity Profile API** is a bit like the Agent Profile API and allows you to store some additional information about an activity in a profile, that can later be queried. It can be used in fairly simple things like leaderboards, or more complex things like language maps.



INTEROPERABILITY AND xAPI

xAPI gives us a structure by which we can communicate - a syntactic set of rules that specifies how we're going to talk about data.

We think about this in terms of the triple - the actor / verb / object - and then the context. Some of these terms are very specific. For example, you can only identify an actor in very specific ways; you must have an IRI (Internationalised Resource Identifier): to identify a verb.

But this doesn't mean we're talking about the same things, just that we're using the same structure. Indeed, sometimes we can use the structure in a conformant manner but end up tracking two different data points in different ways.

In order for us to actually use xAPI it is not enough to adopt the structure, we also have to think a little bit more about the semantics of what we're going to put in the spaces that the structure provides.

If we don't do this we run into some big problems.

For example, the verb 'run' can mean many things - you can 'run' a workshop, or you can 'run' a business, or a marathon. We have to think about how we will use these semantics consistently, not just in our department, but throughout

our business and even throughout our industry.

To truly realise interoperability, we must be talking the same language across organisational boundaries.

Fortunately the writers of the xAPI specification already realised this. To help us define not just how we shape data, but what we say with it, we in the xAPI community refer to the idea of 'controlled vocabulary', 'recipes' and 'profiles' to establish these interoperable standards.

A controlled vocabulary is a published set of xAPI terms (typically the verb) that we might use to describe a particular action. For example, if we want to describe the way in which a user interacts with a video, then we can extrapolate those terms to be relevant to any occurrence of someone using a video online - play, pause, finish; the actions are the same.

The exact same set of terms can be used for every video player, meaning that data from one video Activity Provider is identical to data from another video Activity Provider.

Communities of xAPI adopters have got together to publish their sets of Controlled Vocabulary which you can copy. This means that we all generally define 'playing' a video with the definition held at <http://adlnet.gov/expapi/verbs/played>.

A Recipe takes the concept of Controlled Vocabulary further, giving you a complete set of rules to encode a common activity. This goes further than the vocab and into the structure, objects, properties and the required values associated with each part of an xAPI statement. For example, a Recipe for having 'watched' a video might encode the length of the video, such that we can always tell how long someone spent watching the video in order to trigger the 'watched' statement.

Finally, we can aggregate Recipes together to form what are known as Profiles. Profiles detail a complete set of statements for a particular context. For example, our video player profile could specify that we make at least two statements from a video player; one at the start and one at the end. And so, just as you might have a Recipe for video playback, you might have one for summative assessment, or for social learning, or for tracking a mentoring session.

The community has come together to make profiles for common contexts such as SCORM content and CMI-5 (a potential SCORM successor). Where no relevant community profile is present already, it's suggested that you publish your own and try to explain the rationale behind your decisions for others to follow.



A companion specification that just describes xAPI vocab is available online:
<https://adl.gitbooks.io/>

Creating An Interoperable Approach To Documenting Social Learning Experiences

HT2 Labs' first dealings with xAPI came as a result of building our social learning platform, Curatr.

Using Curatr, learners were tasked with viewing content, making comments and then adding back their own contributions to an online learning portal. Whilst we could track all this inside the system, we couldn't readily export the data to any other platform. This is when we discovered the xAPI and first set about adopting the specification as a means of reporting on what learners had achieved in a social learning platform.

Since deciding to go with xAPI, one of the struggles we've had to deal with is creating the right Recipe for tracking social learning interactions using xAPI. For example, consider how we track a learner 'liking' another user's comment. We could construct a statement that says 'User A liked User B', with the comment itself in the context. That allows us to easily return how many times User B has made comments that have been 'liked'.

But, semantically, it's not true. User A didn't 'like' User B, they 'liked' a particular comment. In which case we would say 'User A liked User B's comment'. That's more accurate, semantically, but now it's harder to count how many 'likes' User B has got, because we're more focussed on tracking how many likes User B got for that particular comment, which is unique, not every comment they contributed.

This is just one example of how adopting the specification has been somewhat harder than we first thought it would be.

WHAT OTHER OPTIONS DO I HAVE, THAT AREN'T xAPI?

We've come this far in life without xAPI so surely there must be alternatives to collecting all this data?

There are a few competing specifications that you might come across and there are different techniques you can use to transfer data, but xAPI is generally in a class of its own. The only other specification on the scene is IMS' Caliper specification. This accomplishes a lot of the same things xAPI does and even looks similar. However, it is limited by what are known as its 'Metric Profiles' - these are a form of recipe that are baked into the specification. You can only use Caliper with the few Metric Profiles that are to be found in the specification itself - covering learning activities like taking tests, watching learning activities and so on.

IMS has taken a 'top down' approach to dictating what you can talk about using Caliper, whereas xAPI is more crowd sourced, a bottom-up approach. That isn't to say Caliper isn't useful; it is. But it's more likely to become a set of Recipes for xAPI than it is to supercede xAPI. In fact IMS and ADL, the two parent organisations involved, are actually working together to make this happen.

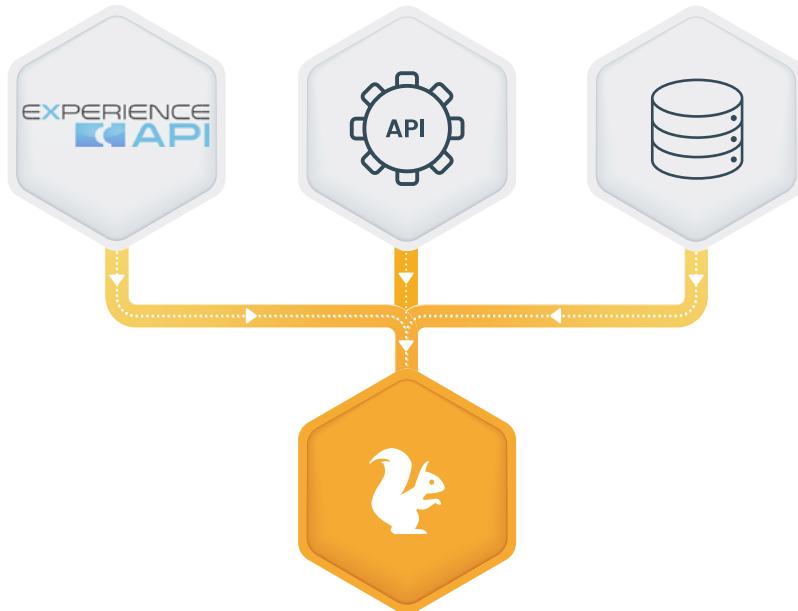
Then there are other ways of

talking between systems. We already talked about API's generally speaking and this is a potentially viable alternative to using xAPI. Where you only need to connect two systems together in a fairly trivial manner

it is entirely appropriate to use a custom API. These will all be somewhat proprietary - whilst you can follow the documentation to connect the two systems they will very rarely work 'out-of-the-box' together, you'll have to customise elements to get them talking. And when you send data over an API between two systems you are in a 'fire and forget' environment. Once the message is sent, it is received but never stored. That makes it hard for a 3rd party to become involved in the relationship. Using an LRS with the xAPI is an easier way to connect multiple systems together using an already standard approach.

WHERE CAN I GET xAPI DATA?

There are a number of ways to start collecting xAPI data. You can integrate software applications that already come prepared to output xAPI, you can customise the output of systems with APIs to translate their language into xAPI, or you can find ways to convert existing databases or spreadsheet data into xAPI.



EXISTING APPLICATIONS

There are more than 150 applications available in the world that already export data in the xAPI format. These include Learning Management Systems (like Moodle, Blackboard or Totara), Content Management Systems (Wordpress and Drupal for example), Authoring Tools (like Storyline, Captivate and Lectora), Social Learning systems (like Curatr), Mobile apps (like Tapestry or Learning Locker's Acorn app), Performance Support tools (like Red Panda, the Trek Learning experience system, or xAPI Apps), Assessment platforms (like TAO Testing or Question Mark) and many more.

It is becoming increasingly common to see 'xAPI enablement' as a key feature in RFP's put out by large organisations.

APIs AND WEBHOOKS

The majority of newer software tools, especially SaaS-based offerings, come with some form of Web Service or API. This allows you to setup scripts that will go and retrieve data from the application for use in your own. It's entirely possible to convert these API's into xAPI format, given that the right data is available. Sometimes you need to set up scripts or triggers to access the API but more modern applications implement WebHooks. These bypass the need to query the API, instead they push data out directly to listening endpoints when an event occurs. Popular tools like Slack, SalesForce and Dropbox all implement WebHooks, as do integration technologies like Zapier.



Zapier is worth a note on its own. This tool and others like it (consider IFTTT) allow you to transform data from one API into another - like xAPI - without having big technical knowledge. They make money based on how much data you put through the services, but as a route to a proof of concept they are well worth investigating.

DATABASES AND CSV

Finally, if no Web Service is available but you can still get access to raw data, all is not lost.

It will take a bit of custom development, but it is possible to convert data from a spreadsheet or SQL database into xAPI format, given a bit of time, the right recipe and access to a developer.

Typically developers create scripts that can Extract, Transform and Load data from these sources into an LRS. In fact there's a whole industry dedicated to the 'ETL' process.

Typically scripts don't run in real-time; they might be used initially as a means of loading data, or might run as a timed event (often referred to as a 'Cronjob') every week or overnight.

HOW IS xAPI USED?

There are many use cases for xAPI but it is most common to see the specification used in connecting systems together, performing data analysis and facilitating performance support.

Typically, when you have to connect multiple systems together, you end up doing a fair amount of custom work to get them talking to each other.

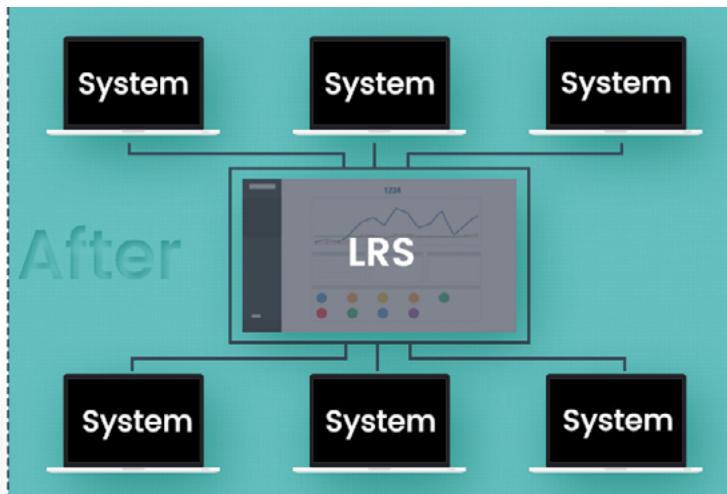
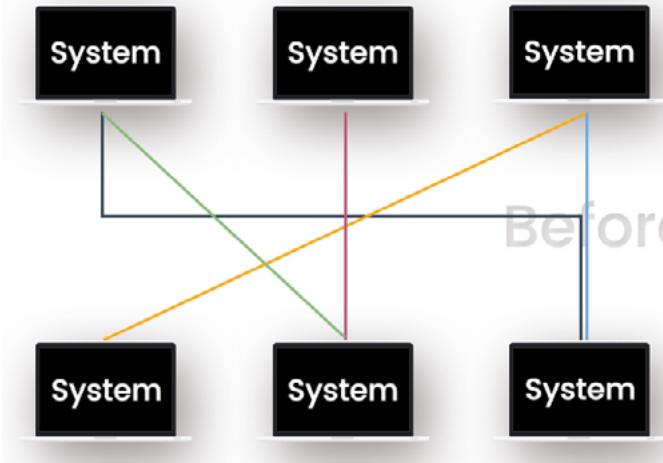
of custom integrations you need to make, it also gives you a single source of record for all this activity - the Learning Record Store.

Sales folks for software companies will tell you that their technology will integrate seamlessly because 'it has an API'. But having just any API is not enough - unless it is a 'standard' API, you will still have to write custom integration methods to get systems talking.

There are many relevant standards to be considered. For example, Single Sign On has a whole raft of 'standard' approaches such as SAML or oAuth. But for sharing learning activity data, there is really only one viable API that has been standardised - xAPI.

In time this will give you more flexibility and choice in the systems you procure. Right now it can be a very demanding task to migrate between Learning Management Systems. Much of this complexity is caused by learning records being kept inside the LMS, in some proprietary format. By adopting a Learning Record Store you take control of your learning data and keep it separate from any system you might want to replace in future.

And because Learning Record Stores are interoperable with each other, you can switch out your LRS provider with a minimum of fuss. This is what makes an LRS so different from an LMS - there is a very specific set of rules that account for what an LRS must do. As long as your LRS adheres to these rules, you are safe.



ANALYSIS AND BENCHMARKING

Presuming you've got some level of connectivity between systems, the next thing you'll want to do is analyse how learning takes place across these various systems. Using this data you can draw some conclusions about which forms of training actually impact performance and which learning behaviours are demonstrated by your highest performers.

To do that, you need to track both the training that is viewed and consumed, and the performance data. This is why xAPI should be of interest to people even outside of the L&D department: the L&D team will need access to this performance data if they are ever going to be able to test their hypotheses and to design better learning.

Analysis in this style isn't straightforward and it's not immediately rewarding. You are stepping towards the world of scientific experiments; attempting to understand the relationship between two variables. And most scientific studies find absolutely no relationship whatsoever!

Science is rigorous. Which also makes it ponderous. If you are wondering why there aren't hundreds of successful case studies from xAPI that show how training impacts performance, it is

not because xAPI doesn't work. xAPI is simple. It's just the way we communicate. But actually finding a relationship between two variables in a controlled, repeatable manner is darn near impossible.

Just because it's hard, doesn't mean you don't have to try. Imagine using that as an excuse to the board in this age of analytics. "Yeah, well I was going to prove the impact of my training, but it turns out it's too hard". That's a sure fire way to get your budget cut...

Finally, it's worth a note about the value of data today for analysis tomorrow. Even if you do not have the ability to analyse data that you are collecting today, you may develop that capability tomorrow. Having a base in historical data will give your analysis more depth and give the opportunity to benchmark current performance against past performance. That's why we'd recommend organisations get started with xAPI sooner, rather than later. It will take time to build up a meaningful amount of data for use in later analysis.

PERFORMANCE SUPPORT

Whilst you might not get many 'Holy Grail' answers from analysis you will probably get a whole heap of pointers on what works better, and again, xAPI can help here. If you've connected your systems together and done the analysis to understand that when a person does X, they may need Y, you can trigger things to happen automatically.

Imagine sending a customised email with revision advice to a person failing an assessment 3 times in a row. Or sending the latest product video out to a sales person the day before they have a meeting on the new product. Or texting new installation advice out to your on-site engineer right as they turn up on-site. Commercial sites like IFTTT and Zapier have become huge working on just these principles. Now you can do it within the confines of your own organisation.

The idea of performing 'just in time' interventions whereby the system identifies that you need help right when you need it has been prevalent in the learning industry for a number of years. As we increasingly hear about the role of AI and Machine Learning in technology, you should know that it will be through the standardised use of data that these techniques are actually brought to life. It's always dull behind the sexiness :)

As a most basic example of Machine Learning, consider a performance support system that not only sends out automated interventions but learns when and how those interventions make a difference to outcomes. If the system sends out help emails and the learner goes on to pass the assessment, it reinforces that the email is a good intervention. If the emails have no effect on the later outcome, then the system learns that this approach has little actual benefit and can try another approach. Right now it might seem like Machine Learning is way away. But xAPI is shaping our ability to exploit these techniques in the near future.

15

INTRODUCING THE LRS

A Learning record Store (LRS) is a database that stores and retrieves data produced in the xAPI format. In an ecosystem where training portals, mobiles, LMSs and other technologies are all submitting “statements” on a moment-by-moment basis, the LRS is a key piece of infrastructure that requires high availability and high reliability.



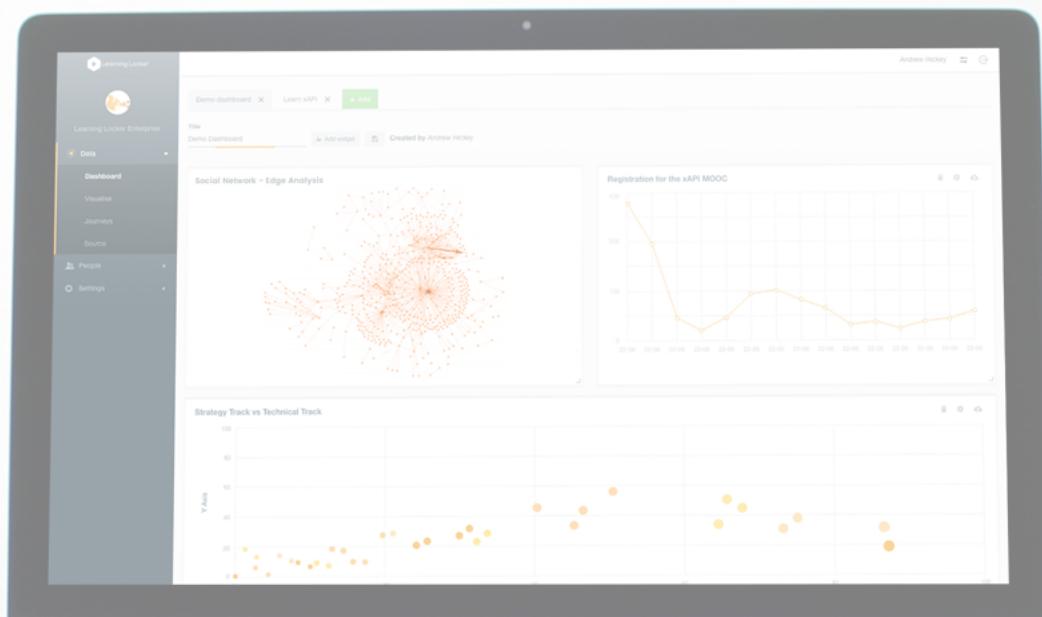
WHEN DO I NEED AN LRS? AM I GETTING RID OF MY LMS?

It's important to note that the term "LRS" is quite specific. Where "Learning Management System" (LMS) can be used to refer to a range of vastly different products, LRS refers to a distinct piece of software that adheres strictly to the xAPI specification for an LRS. There is essentially a recipe card for building an LRS; if it omits part of the spec or tweaks it in some non-standard way, it's not an LRS.

You won't need an LRS until you've got at least one system producing xAPI statements. But as soon as you start producing xAPI statements, you'll need an LRS to store them. The good news is you don't need to get this right first time. LRSs are designed to be interoperable. You can run more than one at once and share data, or readily transfer data out from one to another, presuming that the data in the LRS is valid. There's only really one fatal mistake you can make in getting started and that's in the quality of the data you collect; garbage in, garbage out.

There are a number of fundamental differences between an LMS and an LRS that mean you won't be getting rid of your LMS anytime soon. An LMS will manage access to content, giving you the ability to upload packages of learning content, assigning them to learners and then tracking their progress. An LRS will do none of that - most learners will never even be aware that the LRS is operating in the background, let alone log into it. Your LMS will also probably deal with launching eLearning content packages. This is not part of the xAPI specification and doesn't form part of any standard LRS.

Building an LRS is far from trivial. There are more than 1,500 tests for an LRS to pass flawlessly before it can be considered a 'conformant' LRS. Being conformant is important because one of the absolute key functions of the Learning Record Store is to check that data is valid according to the xAPI specification. And there are a lot of nuances in doing this: If your LRS does not do this to the letter, then you may be storing data that you cannot use again.



SHOULD MY LRS BE PART OF MY LMS?

No one wants to buy a new piece of infrastructure; it's a difficult decision that will have time and costs associated with it. It's attractive to think that an existing system can be upgraded to include the new functionality—for instance, adding an LRS component to your LMS. However, your LMS is more likely to be useful as an Activity Provider (in fact, it's vital that it does this so that you can track how learners use the LMS as well as content). But you will increasingly need to think about a standalone LRS as you think about three key issues: Connectivity, Analytics and Scale.

ANALYTICS

Although analytics are not a part of the xAPI standard per se, they are the reason you might want an LRS. The part of the system that stores data is what allows it to conform to the specification; the ability to interpret this data is what makes an LRS a useful piece of software.

An LMS is fundamentally not built around data analysis; it is built around courses and content. It's not enough to have your LMS "store" xAPI data. That's just the start. You are going to need some really powerful analytics to start answering performance questions. And much of that data might not be the sort of results content driven LMS's are accustomed to dealing in.

CONNECTIVITY

You don't need to be interacting on the LRS to insert data. In fact, most users will never even see the LRS. They won't have accounts; they won't login. Most LMS setups require users to access content and resources through the LMS. But this is a huge barrier to measuring performance – we don't login to the LMS to do our day-to-day work and this is where the xAPI can really excel.

An external LRS will come complete with features like OAuth, a standard that allows external applications to connect to data within a system and to insert new data from external systems. Many LMSs are also setup to charge based on the number of users submitting data to the system. With xAPI there is no limit to the number of users on which data can be collected; you may find yourself in for a big surprise when your next LMS invoice comes around.

SCALE

LRSs will need to store and process vast amounts of data. Where we used to get a handful of data points from a learner to the LMS, we've seen xAPI activities generate 1,000,000 statements in a day following use by a company at scale. This is an order of magnitude larger than the LMS generally deals with and dictates a different technology stack. Typically this means a NoSQL data store - something no mainstream LMS currently available is built upon.

WHAT SHOULD I BE CONCERNED ABOUT WHEN PROCURING AN LRS?

When we first start working with a new client we run through a checklist of questions to help us get the context for an LRS installation. What follows here is a few highlights from that checklist that will give you some key questions you'll want to think about when first procuring an LRS.

01

Will the LRS be hosted in the cloud or deployed on-premise?

One of the first decisions you'll need to make is whether you want to host your own LRS, or use an online service.

An online service will be quicker to setup, probably cheaper short-term (unless your labourer cost is zero), and will be tried and tested. However, you will need to be comfortable with data storage/ownership responsibilities (is all this data OK in the cloud or do you need on-premise?), and you'll need to be comfortable with the medium-long term costs of continually paying for a service.

02

How much data will we create?

It's really easy to imagine that little bits of JSON containing xAPI statements won't add up to much data. And, in the singular, you'd be right. Most xAPI statements are around 2KB in storage size (though we have seen statements ten times that size!). But if you are making a million xAPI statements a day this is going to add up to GB's of data in a matter of moments.

Whilst cloud-based storage is generally very cheap, the technology required to run an LRS at scale does not run well on very basic equipment. For example, the SaaS version of Learning Locker actually runs on 12 virtual machines at a minimum. There is a trade-off between the quantity of data you collect and the level of detail you require in your xAPI statements. Storing more than you absolutely need can be wasteful.

You should be able to estimate your storage requirements by understanding your Activity Providers recipes and multiplying the output by the likely scale of your audience. Getting this number accurate can be important in SaaS circumstances, where you are most often charged on the basis of data stored.

Learning Locker is priced on a 'per GB of stored data', so you pay \$100 (£75) for each GB up to 10GB, then the price per GB decreases. This model allows you to start with a hosted LRS and a modest budget whilst you see how much you use. Then you can always switch to an on-premise deployment if cost is becoming a factor.

03

Are there any location restrictions?

When implementing xAPI you should become hyper-aware of your organisation's data protection privacy policies and the geographic differences that might occur. For example, does your organisation require data to be stored in a particular geographic location? Or, perhaps more likely, are there particular areas of the world you are required to avoid storing your data-at-rest?

04

How is your data secured?

Data security remains a hot topic and your Learning Record Store is no exception to the trend. You should be interested in how your xAPI data is secured whilst in-transit to the LRS and how it is secured at-rest in the database. There is literally no excuse to NOT use SSL whilst sending xAPI data to the LRS. And most cloud providers (including Learning Locker) will offer you encryption-at-rest.

05

How much data can be lost in worst case scenario?

The LRS quickly becomes a key part of your infrastructure. If you are storing all organisational learning data on it then you can't really afford for it to go down, or worse, to lose data. The killer question for redundancy and backup is always how much data can you afford to lose?

Of course the preferred answer is 'none' but that tends to be unrealistic in the face of cost/benefit analysis. In the worst-case scenario,

how much data could you afford to lose and hope to recover normal operating practice? For many circumstances putting in some failover mechanism and also doing off-site daily backups is enough. But in high-risk or testing environments, even that might not be. How can you get that 24hr number down to 1 or 2 hrs without breaking the bank? How much money will you have to spend to reduce this number?

06

And don't forget archiving...

Because xAPI is an immutable specification (you can't edit or delete stored statements) your data set is just going to grow. To keep costs down and keep servers running efficiently, you'll need to develop a process for archiving old data. If you don't the LRS is going to get pretty big in years 2, 3, 4...

HOW TO GET AN OPEN SOURCE LEARNING RECORD STORE

Learning Locker is the Learning Record Store developed by HT2 Labs. Available as both Open Source (for on-premise installation) and Enterprise (for Hosted, Software-as-a-Service), Learning Locker is one of the most popular Learning Record Stores available on the marketplace.

Learning Locker comes ready to accept data in the xAPI format. With your data flowing, Learning Locker allows you to create customisable dashboards using a WYSIWYG interface. Advanced API options are available to sort and share learning data back to Activity Consumers, including the ability to connect data to Business Intelligence tools.

Our Open Source version is a fully featured LRS - in fact our Enterprise version is much the same, apart from the ability to install additional apps. It includes features like:

Customisable Dashboards

Create new dashboards at the click of a button. Use our drag and drop interface to position the visuals you need, where you need them. Export it all at the click of a button

In-built Charting Library

Create new dashboards at the click of a button. Use our drag and drop interface to position the visuals you need, where you need them. Export it all at the click of a button

Trigger Events and Web Hooks

Create new dashboards at the click of a button. Use our drag and drop interface to position the visuals you need, where you need them. Export it all at the click of a button



Installing Open Source

We've made it really simple to get started with Learning Locker on your own hardware.

First, if you use Amazon Web Services, all you need to do is start a new EC2 instance in a location of your choice and, when you go to provision the server, navigate to the 'community AMI' tab. From here you can search for 'Learning Locker'. Once through the AWS setup process you'll have a working Learning Locker instance alive and kicking in about 30 seconds.

If AWS isn't your bag we've still got your back. Simply browse to lrnloc.kr/install and follow the script instructions to install on your own device or another server provider..

Run your own **LRS MOOC**

We've made a complete 4 week course for you to follow to get up and running with your own Learning Locker and push it into production.

Simply visit
www.ht2labs.com/resources/free-lrs-mooc
to get started.

GO ENTERPRISE

Open Source isn't right for everyone. That's why we created our Enterprise, fully managed Learning Locker. Not only does it take the pain away from running an LRS at scale, it also comes with our award-winning customer support and a whole suite of add-on applications that take your Learning Locker to the next level.

Our Enterprise-grade hosting plans come with a 99.5% uptime guarantee, backed up by a full service credit plan. Our fully automated deployment services allow us to spin up Learning Locker anywhere in the world, on any of the major hosting providers, AWS, Google Cloud or Azure. And HT2 Labs is IS027001 and 9001 certified, so your data is in the safest possible hands.

Featured Apps



GDPR App

If you're storing data on EU citizens, then you'll need to comply with the new GDPR regulation. Learning Locker makes this seamless with a plugin application to your LRS that gives individuals access rights to their data through a website that you control.



Business Intelligence Connector

If you're serious about data, chances are your organisation already has a BI tool. That's why we make it easy to stream data in real-time from Learning Locker into tools like Tableau and PowerBI, using our BI Connector App.

Small	Medium	Large
2,000 personas or 20GB	10,000 personas or 50GB	50,000 personas or 100GB
2 Add-on Apps	4 Add-on Apps	8 Add-on Apps
Hosted in EU / US / Aus	Hosted in EU / US / Aus	Single-tenant - Choose Location

Find out more about our Add-on Apps, Pricing and Benefits of Enterprise Learning Locker by visiting

www.ht2labs.com/lrs

DEVELOPING xAPI STATEMENTS

22

This next section will cover the creation of a basic xAPI statement by going into some detail about what each element of a typical xAPI statement may contain. It gets a bit technical in places but it is worth the time trying to understand the structure behind xAPI as it will mean data analysis, and actually using your data, will become a lot easier down the line.

```
        "suspended": false,
        "score": "2016-09-01T11:41:15.482Z",
        "active": true,
        "line_id": "58da679c8c5d017e4004714f",
        "line": {
            "_id": "58da679c8c5d017e4004714f"
        },
        "completedQueues": [
            "STATEMENT_PERSON_QUEUE",
            "STATEMENT_QUERYBUILDERCACHE_QUEUE",
            "STATEMENT_JOURNEY_QUEUE"
        ],
        "statement": {
            "authority": {
                "objectType": "Agent",
                "name": "New Client",
                "mbox": "mailto:hello@learninglocker.net"
            },
            "stored": "2016-09-01T11:41:15.402900+00:00",
            "context": {
                "registration": "a02b534f-5248-4bfd-8512-abbeef63570cf",
                "contextActivities": {
                    "grouping": [
                        {
                            "objectType": "Activity",
                            "id": "http://beta.curatr3.com/courses/elearning-beyond-the-next-button",
                            "definition": {
                                "type": "http://activitystrea.ms/schema/1.0/article",
                                "name": {
                                    "en-GB": "RESEARCH: The Decisive Dozen - Research Background Abridged"
                                },
                                "description": {
                                    "en-GB": "We've used this research from Dr Will Thalheimer in a previous level, but we wanted to resurface it again as elements of spaced learning. Section 10 (page 6) refers to 'Spacing'.\r\n\r\n\r\n(n20 mins+)\r\n\r\n(2 mins for the most pertinent part)"
                                }
                            }
                        },
                        {
                            "objectType": "Activity",
                            "id": "http://beta.curatr3.com/courses/elearning-beyond-the-next-button/learn#level/2848",
                            "definition": {
                                "type": "http://curatr3.com/define/type/level"
                            }
                        },
                        {
                            "objectType": "Activity",
                            "id": "http://beta.curatr3.com",
                            "definition": {
                                "type": "http://curatr3.com/define/type/"
                            }
                        }
                    ]
                }
            }
        }
    }
```

HOW DO I WRITE AN xAPI STATEMENT?

Beyond some very simple proofs of concept you are going to want your systems to ‘write’ your xAPI data. Doing it by hand isn’t a feasible, nor desirable approach. That said, you will still want an understanding of the data being created and you might even need to step in and customise it on occasion. This part of the guide will give you a brief introduction to the Statement API and the variables held within each part of a typical xAPI statement, starting with the Actor, then the Verb, then the Object and then, everything else!

UNDERSTANDING THE ACTOR ELEMENT ON AN xAPI STATEMENT

So, first things first - an actor can be one of two things: an Agent or a Group. An Agent represents a singular person and a Group represents... well, a group of people. You’re probably going to use an Agent for the most part. An Agent can be identified in 4 different ways: mbox, mbox_sha1sum, openid, or an account. The most robust tends to be using an account.

Agent Actors

Agents can be identified in 4 different ways

The mbox is the user's email address

```
{  
  "objectType": "Agent",  
  "name": "John Smith",  
  "mbox": "mailto:johnsmith@example.com"  
}
```

mbox_sha1sum is an encrypted version of the email address

```
{  
  "objectType": "Agent",  
  "name": "John Smith",  
  "mbox_sha1sum": "4445904ac65039ef7a91506207f19162ac4dea73"  
}
```

The openid is a URL

```
{  
  "objectType": "Agent",  
  "name": "John Smith",  
  "openid": "http://www.example.com/johnsmith"  
}
```

The Account type is a unique identifier, plus a Homepage URL

```
{  
  "objectType": "Agent",  
  "name": "John Smith",  
  "account": {  
    "name": "123",  
    "homePage": "http://www.example.com/users/"  
  }  
}
```

Group Actors

Group actors can be either identified or anonymous

As with Agent Actors, a Group can also be identified by the above 4 elements - in which case it is an 'Identified Group'

```
{  
  "objectType": "Group",  
  "name": "HT2",  
  "account": {  
    "name": "123",  
    "homePage": "http://www.example.com/users/"  
  }  
  "member": []  
}
```

Both group types can have members - if the group is not an 'Identified Group', it is an 'Anonymous Group' and must have at least 1 member

```
{  
  "objectType": "Group",  
  "name": "HT2",  
  "member": [{  
    "objectType": "Agent",  
    "name": "John Smith",  
    "account": {  
      "name": "123",  
      "homePage": "http://www.example.com/users/"  
    }  
  }]  
}
```

If you're wondering what a complete Actor statement looks like, here it is

```
{  
  "actor": {  
    "objectType": "Agent",  
    "name": "John Smith",  
    "account": {  
      "name": "123",  
      "homePage": "http://www.example.com/users/"  
    }  
  },  
  "verb": {  
    "id": "http://adlnet.gov/expapi/verbs/completed",  
    "display": {  
      "en-GB": "completed"  
    }  
  },  
  "object": {  
    "objectType": "Activity",  
    "id": "http://www.example.com/activities/1"  
  }  
}
```

MANAGING MULTIPLE ID'S

One significant challenge to be overcome when connecting multiple systems together is reconciling user ID. It is increasingly common to see Single Sign-On (SSO) systems in place that make each user within your organisation easy to identify, no matter which system they are logged on. But, where SSO isn't so prevalent, or where each user isn't issued a unique login, or where you are integrating systems outside the organisation's jurisdiction, then you may find yourself storing data for the same user under two completely different IDs. Sometimes this means having an mbox and an account for the same person (and sometimes even multiple mbboxes).

Learning Locker can automatically reconcile multiple IDs into the same persona for simple querying. This feature is a potentially very important requirement if multiple IDs are likely to be a reality in your organisation.

When a new statement is submitted into Learning Locker, the system starts a worker in the background to try and figure out if we've seen this actor before. Sometimes that's really straightforward, like when the same account ID is used.

If that's not the case Learning Locker will check on the other metadata available from the new statement (further personally identifiable information like name or a grouping) and try to match it against existing users. You can configure Learning Locker to be more or less rigorous in this process using a scoring mechanism.

If the scoring mechanism finds a match, Learning Locker will merge the two actors into the same persona. This process doesn't modify any xAPI data, it simply adds a reference within the system. If the scoring mechanism indicates that the new user is similar, but not the same as an existing user, you'll be prompted to make a manual reconciliation. Then any remaining new users that can't be found will automatically be created as a 'new' user within Learning Locker.

Problem solved!

What is an IRI? Or a URI?

At various points in this document you will see us describe an identifier as a URL, or an 'IRI', or maybe even a 'URI'. A URL you probably know. That's an Internet address. But what about URI and IRI?

URL - Uniform Resource Location. This is the link at which a resource resides on a network. This is what you are most likely to be referring to when using a link on the world wide web: <http://www.ht2labs.com> is a URL for example.

URI - Uniform Resource Identifier. A unique identifier for a resource that can be on a network or somewhere else. A URL is actually a type of URI, where the resource in question is identified by its network access. An ISBN number is a URI designed to name a book in a unique manner and at worldwide level.

IRI - Internationalized Resource Identifier - is a generalization of the URI. While URI supports only ASCII encoding, IRI fully supports international characters

So why bother with these elements? It's all about being able to reference something uniquely. Like the way only one person in the world can have your mobile phone number, only one location in the world can be described by a URL. That makes whatever resides at the end of that address unique.

For example, we use a HomePage as well as an Account ID when using the 'Account' actor type. How come? Well, whilst your user ID (123456) might be unique in your organisation, there could easily be another 123456 in another company.

But if we add your unique ID to a homepage URL that represents where you come from (E.G. <http://www.ht2labs.com>), then we can refer to you as user 123456@http://www.ht2labs.com. And there's only one of those!

xAPI makes use of IRI's (which are nearly always URLs in practice) for the identification of verbs, activities and more. Whilst it is best practice that a Verb IRI resolves to some explanation of the verb meaning, it isn't actually required to work to be spec conformant.

UNDERSTANDING THE VERB ELEMENT OF AN xAPI STATEMENT

Verbs get a lot of attention within the xAPI world. Whilst they are the ‘headline’ of what someone did, they don’t carry a lot of detail as to the context in which they did it. So whilst Verbs are important, they aren’t very complex.

A verb can contain two properties - an ID, which uniquely identifies the verb and its meaning, and the Display - which tells the LRS how to display it in a human-readable format (and can be mapped into various languages for translation).

The ID qualifies the meaning of the verb being used (e.g. do

we mean John ‘fired’ a gun, or that he ‘fired’ an employee). We do this by using a unique IRI to describe the Verb. It’s important to note that an IRI does not need to resolve to be unique (i.e. it doesn’t need to exist as a live Web address). However, it is best to use a resolving link that displays more detail as to the Verb definition where possible.

There are no ‘official’ verbs in xAPI, you can use whatever you want. But you don’t need to reinvent the wheel when choosing which ones to use in your statements - in fact, you are much better off using existing verbs where possible.

A number of resources come in handy to check against Verbs that other people have already used; see the list at the bottom of this page.

If a verb that you wish to use isn’t currently listed in the registry you use we recommend that you ask for it to be created. You can make a new IRI to define your Verb’s meaning, just try not to re-invent what others have done before.

An Example Actor Verb Summary

There is one exception to the lack of ‘official’ verbs - that’s the verb ‘Voided’. This is a special, reserved Verb that has a particular meaning. When the LRS receives a ‘Voided’ statement it will look to cancel out the Object referenced. In this way we can remove erroneous statements from being returned by the LRS. Any statement that has been voided will behave as if it was never created. This isn’t the same as deletion; the original statement still exists, it just never gets found when searching.

```
{  
  "actor": {  
    "objectType": "Agent",  
    "name": "John Smith",  
    "account": {  
      "name": "123",  
      "homePage": "http://www.example.com/users/"  
    }  
  },  
  "verb": {  
    "id": "http://adlnet.gov/expapi/verbs/completed",  
    "display": {  
      "en-GB": "completed"  
    }  
  },  
}
```

ADL’s Controlled Vocabulary (<http://xapi.vocab.pub/datasets/adl/>)
Tin Can Registry (<https://registry.tincanapi.com/>)
xAPI Quarterly’s xAPI Index (<http://xapiquarterly.com/the-xapi-index/>)

UNDERSTANDING THE OBJECT ELEMENT OF AN xAPI STATEMENT

In the xAPI world an object can be four things:

- Activity (e.g. something that you do/did)
- Agent/group (e.g. an individual, or group of people)
- Sub-statement (e.g. a nesting of statements)
- Statement Ref (e.g. a reference to another statement, normally used to void an existing statement, or refer to a statement that is yet to exist).

A simple object statement looks like the one opposite; an Activity that has an ID which is an IRI. In this case we're saying John Smith completed <http://www.example.com/>

activities/1.

If you set your Object as an Activity you have to declare that it has an ID (and that it is an Activity). You can extend the meaning further using Activity Definitions (see next page for an example). These definitions make your Statement more useful in later querying. In particular Name and Type can come in useful, as these allow you to group 'like' activities together.

Your Object doesn't have to be an Activity. It is perfectly acceptable for your statement to be about people instead of activities. For example, if John Smith 'followed' Ben (in the Twitter sense), you could set the Object type to 'Agent' instead of 'Activity' and identify Ben in the Object space.

Finally, your Object can also be about other xAPI statements. For example, when you Void a previously submitted Statement you use the 'Statement Ref' to refer to the previous Statement that you wish to Void (using the unique Statement ID that was created when the xAPI Statement was first recorded in the LRS).

An Example Object where Type is Activity

```
{  
  "actor": {  
    "objectType": "Agent",  
    "name": "John Smith",  
    "account": {  
      "name": "123",  
      "homePage": "http://www.example.com/users/"  
    }  
  },  
  "verb": {  
    "id": "http://adlnet.gov/expapi/verbs/completed",  
    "display": {  
      "en-GB": "completed"  
    }  
  },  
  "object": {  
    "objectType": "Activity",  
    "id": "http://www.example.com/activities/1"  
  }  
}
```

Activity Definitions

An Object requires an ID (which is an IRI), an object type and, optionally, further definition.

If you're looking to do more intensive querying on your LRS data, you should set the definition property.

This further definition can have a number of properties:

- **Name** - describes the activity in short, readable format.
- **Description** describes the activity in more detail; again human readable.
- **Type** (IRI) - suggests what sort of Activity this Object was, e.g. a video, an assessment, a simulation, etc.
- **More Info** (IRL) - a link to further information about the activity, which is useful if the description is long.
- **Extensions** - A series of further extension that can be used to better identify the Activity or where it took place, for example longitude and latitude coordinates of where an activity physically took place in the world.
- **Interaction Properties** - Tend to be used for describing SCORM-like data, such as question response patterns.

The key thing to be aware of when programming your Object statements is that although only the ID is required, you really do need to go into detail when setting your definitions. Without this level of detail you won't be able to filter your statements back out of your LRS data.

This process we often refer to as 'designing for data'. For example, if you want to know who has watched the most videos it is far easier to lookup where 'Activity Type = Video' than it is to look up every single Object ID that happens to be a video. And if you wanted to know the physical location of your learners you would need to be storing the Long/Lat co-ordinates at the time of the Activity. This data is immutable and can't be changed after-the-fact, so you need to get it right.

An Example Object Statement Summary

```
{  
  "actor": {  
    "objectType": "Agent",  
    "name": "John Smith",  
    "account": {  
      "name": "123",  
      "homePage": "http://www.example.com/users/"  
    }  
  },  
  "verb": {  
    "id": "http://adlnet.gov/expapi/verbs/completed",  
    "display": {  
      "en-GB": "completed"  
    }  
  },  
  "object": {  
    "objectType": "Activity",  
    "id": "http://www.example.com/activities/1",  
    "definition": {  
      "name": {  
        "en-GB": "completed"  
      },  
      "description": {  
        "en-GB": "completed"  
      },  
      "type": "http://adlnet.gov/expapi/activities/question",  
      "moreInfo": "http://www.example.com/activities/1/more",  
      "extensions": {  
        "http://www.example.com/ext": true  
      },  
      "interactionType": "choice",  
      "correctResponsePattern": "example"  
    }  
  }  
}
```

An Example Result Statement Summary

A complete statement set up correctly with a result will look something like this:

```
{  
  "actor": {  
    "objectType": "Agent",  
    "name": "John Smith",  
    "account": {  
      "name": "123",  
      "homePage": "http://www.example.com/users/"  
    }  
  },  
  "verb": {  
    "id": "http://adlnet.gov/expapi/verbs/completed",  
    "display": {  
      "en-GB": "completed"  
    }  
  },  
  "object": {  
    "objectType": "Activity",  
    "id": "http://www.example.com/activities/1"  
  },  
  "result": {  
    "score": {  
      "scaled": 1,  
      "min": 0,  
      "max": 100,  
      "raw": 100  
    },  
    "success": true,  
    "completion": true,  
    "response": "Example string",  
    "duration": "P1DT12H",  
    "extensions": {  
      "http://www.example.com/ext": true  
    }  
  }  
}
```

UNDERSTANDING THE RESULT ELEMENT OF AN xAPI STATEMENT

Now that we've covered the Actor > Verb > Object triplet, it's time to turn towards other attributes we can add as context. One of the most sought after pieces of context is the Result - a score and associated information that might come as a result of taking a quiz or assessment.

Result statements can include a number of optional details:

- **Score**
 - **scaled** (a percentage between -1 and 1)
 - **raw** (the actual score recorded)
 - **min** (minimum score achievable)
 - **max** (the maximum score achievable).
- **Success** - Pass or Fail.
- **Completion** - Yes or No.
- **Response** - e.g. the actual comment/answer given in response to a question.
- **Duration** - how long it took to complete the activity.
- **Extensions** - e.g. more detail about the result.

Using xAPI you can track the result of each individual question within an assessment or the assessment as a whole. For example, we could submit each question answer as an xAPI Statement and then submit a parent Statement that references earlier question responses. In this manner we can record the overall result as well as a question-by-question analysis.

Later analysis is really quite straightforward using the Result context and xAPI. Now we can quickly evaluate the average score for an assessment, which questions are easiest (or hardest) and how long people spend answering each question.

An Example Result Statement Summary

A complete statement with context may look something like this:

```
{  
  "actor": {  
    "objectType": "Agent",  
    "name": "John Smith",  
    "account": {  
      "name": "123",  
      "homePage": "http://www.example.com/users/"  
    }  
  },  
  "verb": {  
    "id": "http://adlnet.gov/expapi/verbs/completed",  
    "display": {  
      "en-GB": "completed"  
    }  
  },  
  "object": {  
    "objectType": "Activity",  
    "id": "http://www.example.com/activities/1",  
  }  
  "context": {  
    "registration": "957f56b7-1d34-4b01-9408-3ffeb2053b28",  
    "instructor": {  
      "objectType": "Agent",  
      "name": "Joe Bloggs",  
      "account": {  
        "name": "321",  
        "homePage": "http://www.example.com/users/"  
      }  
    },  
    "team": {  
      "objectType": "Group",  
      "name": "HT2",  
      "account": {  
        "name": "123",  
        "homePage": "http://www.example.com/groups/"  
      }  
    },  
    "contextActivities": {  
      "parent": [{  
        "objectType": "Activity",  
        "id": "http://www.example.com/activities/parent"  
      }],  
      "grouping": [{  
        "objectType": "Activity",  
        "id": "http://www.example.com/activities/grouping"  
      }],  
      "category": [{  
        "objectType": "Activity",  
        "id": "http://www.example.com/activities/category"  
      }],  
      "other": [{  
        "objectType": "Activity",  
        "id": "http://www.example.com/activities/other"  
      }]  
    },  
    "revision": "1.0.0",  
    "platform": "Example",  
    "language": "en-GB",  
    "statement": {  
      "objectType": "StatementRef",  
      "id": "957f56b7-1d34-4b01-9408-3ffeb2053b28"  
    },  
    "extensions": {  
      "http://www.example.com/ext": true  
    }  
  }  
}
```

As you can tell, if you include all of this data in your statement then it's going to get pretty long - but if you're looking at really getting the most from your data, then there are a couple of additional elements you might want to consider...

ADDITIONAL PROPERTIES IN YOUR xAPI STATEMENTS

Statement ID

Each xAPI statement stored is given a Unique identifier for the statement - the UUID.

The LRS will set it for you unless you explicitly set it yourself - which some might choose to do so that they can use the ID elsewhere, without querying the LRS for it.

You'll get an error if you create an ID that already exists - so you must be aware of what else exists in the LRS. The UUID is a long string:

```
8caba439-5455-485d-8c99-b3680ef8e46f
```

Timestamps

There are two ISO-formatted timestamp properties in a statement. They look like this:

The “timestamp” property which represents the time that the experience took place;

```
2016-05-27T13:06Z
```

The “stored” property which represents the time that the LRS stored the statement.

The “stored” property is set by the LRS, but you can set the “timestamp” property. You can use this feature to track offline activities that sync with the LRS at a later time. The time at which the event actually took place would be recorded with the ‘timestamp’ property; the time it finally hit the LRS would carry the ‘stored’ property.

If you don't set the “timestamp” property, the LRS will set it to the same timestamp as the “stored” property.

Version

The version of the xAPI that the statement conforms to, which is set to the latest version by the LRS if you don't set it.

```
"1.0.3"
```

Attachments

These might represent certificates or badges that have been sent with the statement to be stored in the LRS. Only the attachment's headers are stored within the statement, such as the content type and length of the attachment, rather than the data itself.

```
{
  usageType: "http://www.example.com",
  display: {},
  description: {},
  contentType: "application/json",
  length: 100,
  sha2: "9cfe7faff7054298ca87557e15a1
0262de8d3eee77827417fbdfa1c41b9ec2
3",
  fileUrl: "http://www.example.com/
files/123"
}
```

A Complete xAPI Statement

So what does this all look like when put together in one statement?

```
{
  "actor": {
    "objectType": "Agent",
    "name": "John Smith",
    "account": {
      "name": "123",
      "homePage": "http://www.example.com/users/"
    }
  },
  "verb": {
    "id": "http://adlnet.gov/expapi/verbs/completed",
    "display": {
      "en-GB": "completed"
    }
  },
  "object": {
    "objectType": "Activity",
    "id": "http://www.example.com/activities/1"
  },
  "id": "8caba439-5455-485d-8c99-b3680ef8e46f",
  "timestamp": "2016-05-27T13:06Z",
  "stored": "2016-05-27T13:06Z",
  "authority": {
    "objectType": "Agent",
    "name": "John Smith",
    "account": {
      "name": "123",
      "homePage": "http://www.example.com/users/"
    }
  },
  "attachments": [
    {
      "usageType": "http://www.example.com",
      "display": {},
      "description": {},
      "contentType": "application/json",
      "length": 100,
      "sha2": "9cfe7faff7054298ca87557e15a10262de8d3eee77827417fbdfa1c41b9ec23",
      "fileUrl": "http://www.example.com/files/123"
    }
  ]
}
```

SENDING xAPI DATA

We know that xAPI is parsed using JSON and sent over a Web Service. But how do you make this process bulletproof? How can you be sure your xAPI data arrived at the LRS? Here's three tips to help you out:

3 Batch your xAPI statements.

If your Activity Provider is sending a lot of statements you may want to investigate 'chunking up' requests to place less load on your LRS. It's generally easier for a server to process 1 request with 1000 statements, than it is to process 1000 requests with 1 statement each. Both methods are perfectly valid; the LRS can differentiate between the observed time and the stored time of statements.

1 Use standard code libraries to send activity data.

The easiest way to be conformant with the specification is to use standard code libraries to generate your Statements. You can find these on Github as linked to by the ADL - there is one for each of the major web programming languages. By using these libraries to create xAPI data in the first place you know it will be valid.

2 Setup a queuing system to send xAPI data.

For most production systems it is not good enough to simply 'fire and forget' xAPI statements; they should be queued and tracked to make sure they actually get to the LRS.

For example, if the LRS wasn't available for any reason, would the Activity Provider cache/store the statement to be resent at a later time? A conformant LRS will notify an Activity Provider when a statement has been successfully stored (returning the statement ID created as a result of new data being stored). This makes it relatively easy for a queuing system to retain an xAPI statement until it is certain the LRS has received, and stored, the data.

A black and white photograph showing two people in an office environment. A man with glasses and a blue shirt is seated at a desk, smiling and looking towards the camera. A woman with long hair is partially visible behind him, also at a desk. In the background, there are large windows and office partitions.

More than
2,500 organisations
have downloaded
and installed
the open source
version of
Learning Locker

EARLY ADOPTERS OF THE xAPI

35

With more than 150 applications directly created to work with xAPI there is no doubt that the adoption of the specification is becoming mainstream throughout businesses, education and government organisations. In this final section we will highlight a few of the use cases we've been lucky enough to work with over the previous two years. We aren't able to name the organisations using xAPI and Learning Locker in every case but rest assured, you know their names!



USING xAPI TO ENABLE GAMIFICATION THROUGHOUT THE LEARNING PROCESS

We worked with a major energy provider to implement a new ‘gamification’ system. The aim of the project was to incentivise sales personnel to engage with further training and then benchmark their sales performance back against their participation in training. Does training make a difference?

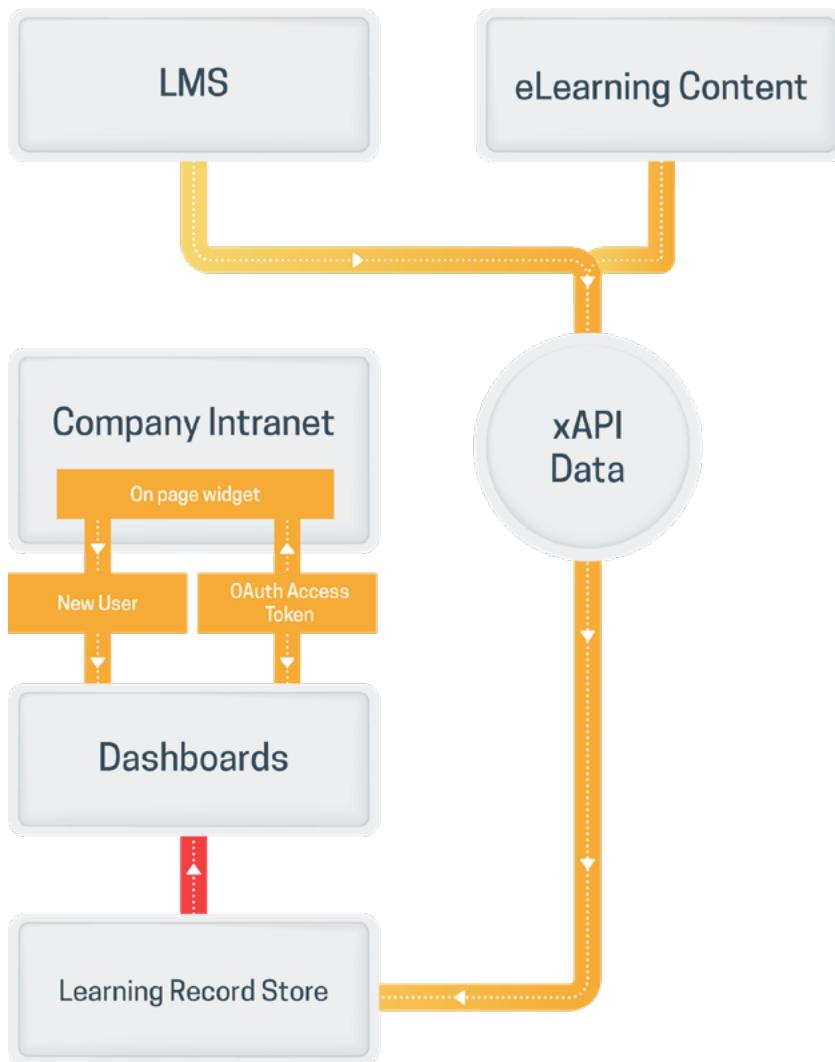
The audience was competitive by nature and responded well to game-like challenges. Product knowledge was spread amongst multiple systems (LMS, Intranet, eLearning content, Apps and more), meaning the gamification ‘engine’ would need to take account of user engagement from all of the available systems.

HT2 Labs implemented Learning Locker to become the central point of data warehousing between the systems and worked with the client and its partners to make Activity Providers export xAPI data based on usage. HT2 Labs

then devised a ‘gamification engine’ that would enable the customer to create challenges based on paths of content to be completed.

As data flowed into the LRS these challenges would then be automatically ‘completed’ based on user interaction. Successful completion of a challenge would lead to ‘digital badge’ being issued to the user.

To show results back to users, a series of ‘widgets’ were constructed including leaderboards and performance graphs. These widgets were integrated with the existing CMS such that administrators could insert gamification elements to highlight performance on any given page. Finally, a series of dashboards were built for the customer to inspect the performance of its distributors, including the ability to benchmark geographic areas against each other.



Activity Provider

Company Intranet
e-Learning Authoring tools
Learning Management System



Learning Record Store

Learning Locker (Enterprise Edition)
running on AWS



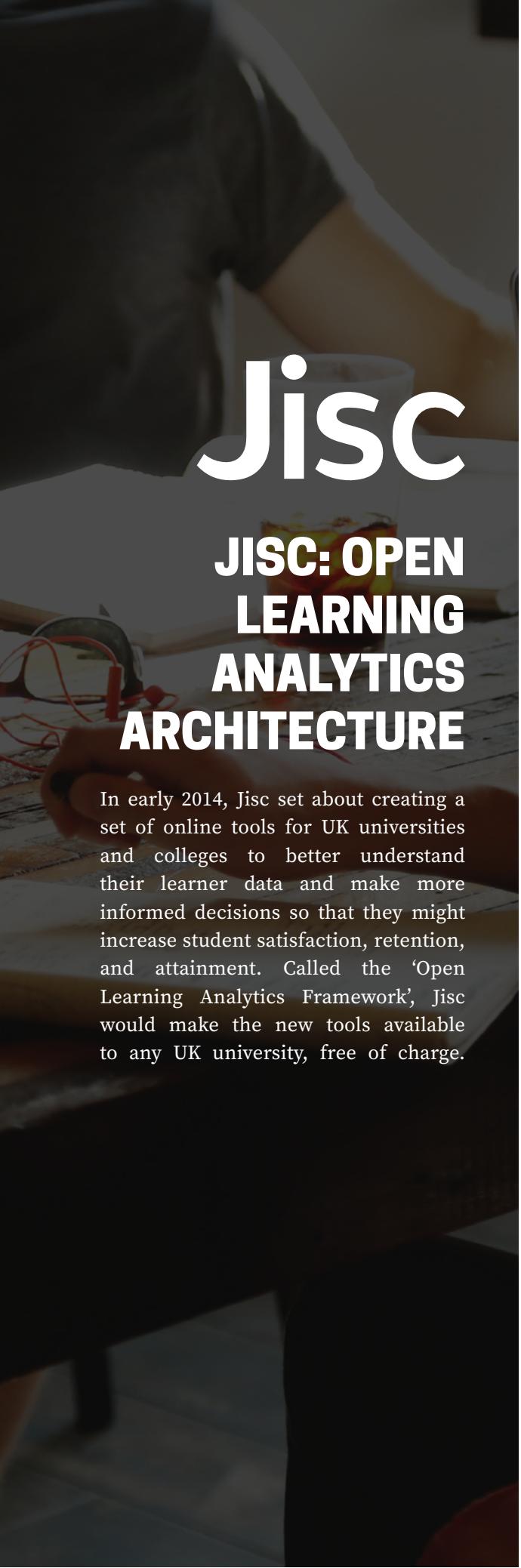
Activity Consumer

Custom Gamification Engine
Custom Dashboards
Company Intranet



Solution

The customer highlighted a need to foster greater engagement amongst distributors with available product knowledge. Research had shown that those distributors whom engaged more with learning materials went on to drive a higher level of overall sales. But how could they drive engagement?



Jisc

JISC: OPEN LEARNING ANALYTICS ARCHITECTURE

In early 2014, Jisc set about creating a set of online tools for UK universities and colleges to better understand their learner data and make more informed decisions so that they might increase student satisfaction, retention, and attainment. Called the 'Open Learning Analytics Framework', Jisc would make the new tools available to any UK university, free of charge.

Research suggests that the use of Learning Analytics by tutors and administrators in Higher Education can improve learning outcomes for students. This includes benefits drawn from 'Student Success' tools that accurately predict whether or not a student is at risk of failure, as well as deeper analysis of learning content and programmes. But the implementation cost of these tools has been a real turn-off for universities looking to engage; there are significant barriers to entry in time and cost. Jisc created a £2m project with the aim of providing a basic set of learning analytics tools, capable of drawing data from a range of sources, at no initial cost to the university.

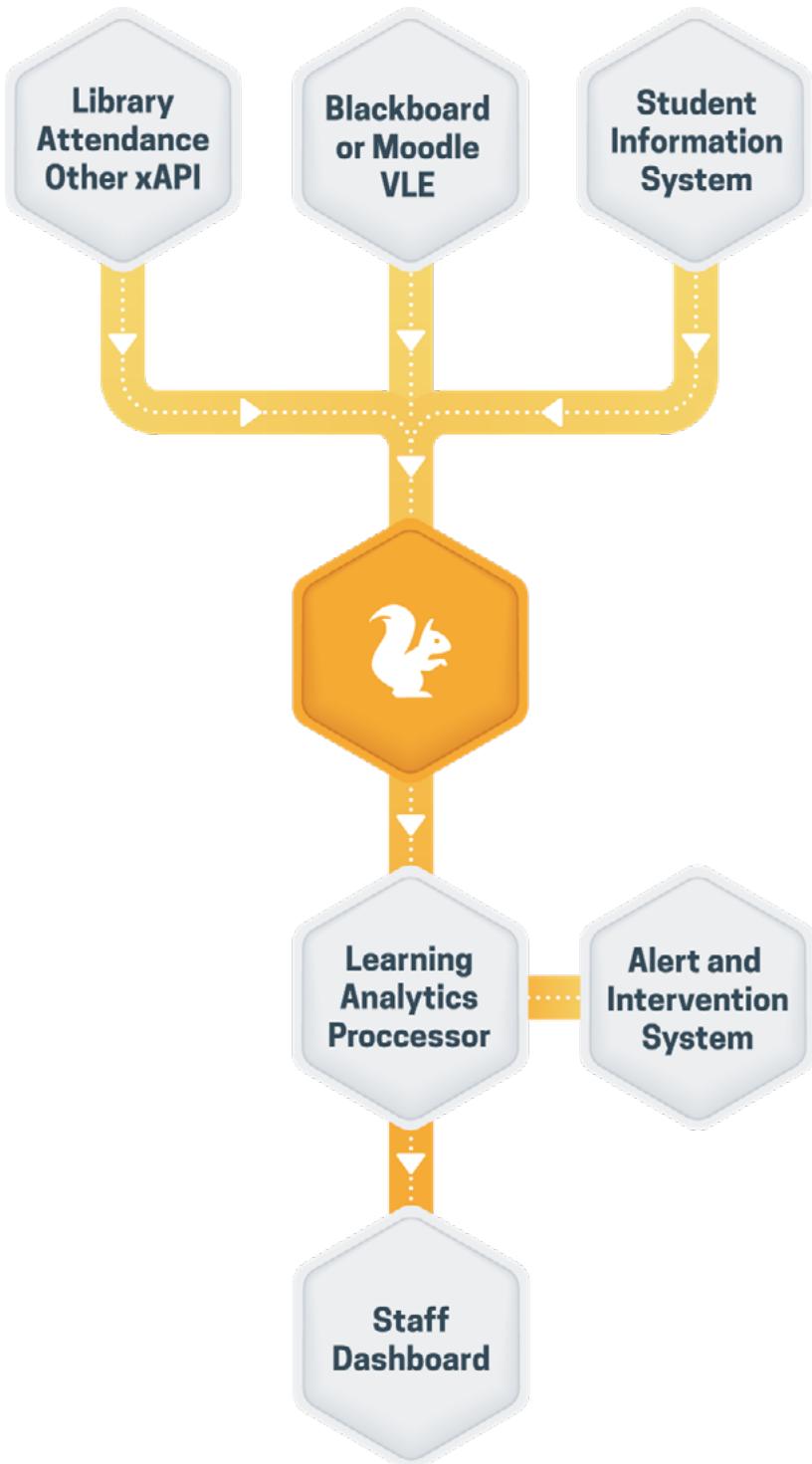
Jisc's solution was to be predicated on 'standardising' the way in which universities collect activity data about their students. Jisc selected the xAPI as the only viable option currently available in education to fulfill this remit. As a result of their decision, Jisc went on to procure a suitable Learning Record Store: Learning Locker was selected due to its Open Source nature and proven ability to scale.

Working as part of a consortium of vendors,

HT2 Labs has implemented Learning Locker for Jisc and advanced its functionality significantly to include additional security features and to expand the record store into a Data Warehouse, capable of storing student data as well as xAPI.

Additionally, HT2 Labs worked with partners to create best-in-class integrations with Learning Management Systems like Moodle and Blackboard. Thanks to this work, LMSs that represent more than 90% of the global install base for Higher Education now directly export their data to xAPI.

Predictive tools like Unicon's Early Alert System and Tribal's Student Success Planner are now capable of taking their data directly from Learning Locker. Jisc have succeeded in creating an Open Learning Analytics service, to which universities can be quickly subscribed and start sending in data without the need for any coding or complex integration time.



Activity Provider

Learning Management Systems
Library Data
Attendance Systems



Learning Record Store

Learning Locker (Enterprise Edition)



Activity Consumer

Unicon Early Alert System
Tribal Student Success
Civitas Learning
Student Mobile App
Tableau



Solution

Today, Jisc is focussing on creating the UK's national learning record warehouse, capable of storing activity data for millions of students with billions of xAPI statements stored. The eventual aim of this data collection is to offer national benchmarking services to UK universities; allowing tutors and administrators to assess the quality of their learning offering when compared to the national average for similar programmes.



Activity Provider

Curatr
SAP



Learning Record Store

Learning Locker (Open Source Edition)



Activity Consumer

Custom Dashboards



Solution

V&B had a strategic requirement to develop a new digital learning capability to reduce time to competence and increase sales effectiveness of their global retail workforce (circa 12,000 employees). With no internal capacity and no budget to outsource content development, V&B needed to tap into the existing knowledge held by their remaining workforce - by crowd sourcing content and best practices

from the people on the sales floor.

V&B selected HT2 Labs' Social Learning Platform Curatr as the software to deliver on these requirements. Using Curatr, the L&D team assembled courses quickly using any sort of digital asset as content - video, PDF's, presentations or even other web-pages. By linking these assets together to form 'levels', V&B encourages learners to view content and contribute to discussions in order to unlock content 'at the next level' - a gamified approach to the learning experience. Comments are welcome on nearly every item of content and discussion points

are raised to get the conversation started. The better learners contribute to the conversation, the more points they earn.

Curatr is capable of exporting all learning activity data directly to an LRS using the xAPI. Here data is stored globally for V&B in a format that can then be used in more complex analysis and reporting. By blending this data from Learning Locker with insights as to sales performance taken from SAP (the global ERP system for V&B), the company was able to demonstrate clear links between the uptake of training and subsequent regional performance increases.



Villeroy & Boch

1748

PROVING RETURN ON INVESTMENT WITH xAPI DATA

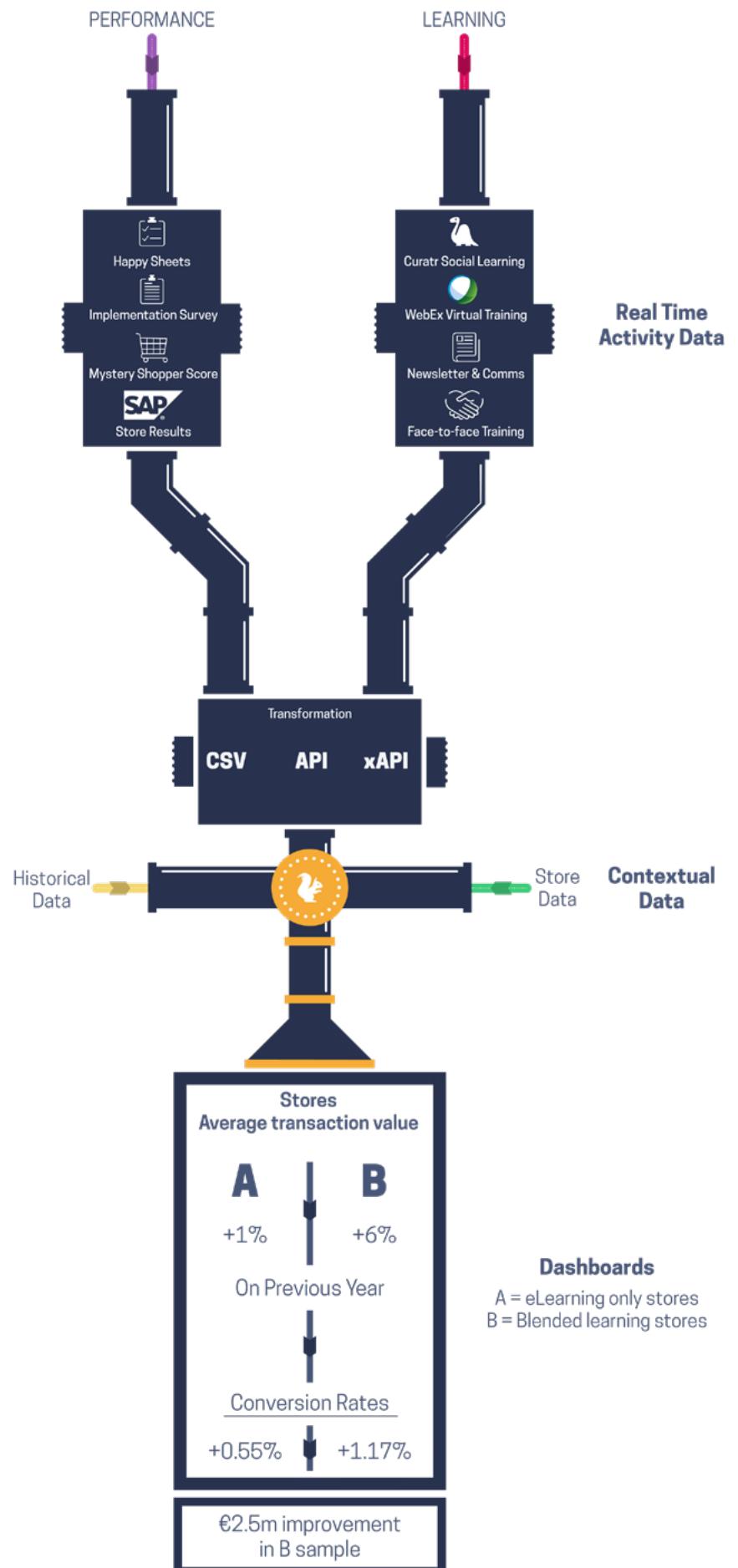
Winners of the Best Social Learning project at the Learning & Performance Institute Awards, 2016, V&B were able to use the data collected by xAPI to prove the link between training and performance throughout their retail stores.



**SOCIAL AND COLLABORATIVE
LEARNING AWARD**

2016 GOLD WINNER





The following figures demonstrate just part of the Return on Investment seen:

- Average Transaction Value is +32% since rolling out 'More Excellence in Selling' (which has a focus on upselling) to UK and USA
- Since rolling out the basic 'Excellence in Selling' in Iberia in May 2015, sales are +15%
- In Japan, 'Porcelain Product Knowledge' was rolled out in June 2015. Sales in this product category are +19% since that time



CITY & GUILDS TECHBAC® FROM EDUCATION TO EMPLOYMENT USING THE xAPI

This project won Best in Show at DevLearn 2015 for its ingenious use of xAPI to connect a series of learning systems together seamlessly.



BEST OF SHOW VENDOR
2015



Momentum quadrant





Activity Provider

Custom Learning Portal
CV Builder
3rd party applications



Learning Record Store

Learning Locker (Enterprise Edition)



Activity Consumer

Custom Dashboards
Mozilla Open Badges



Solution

City & Guilds TechBac® is an entirely new professional programme designed to give 14 to 19 year olds an exciting alternative path towards an apprenticeship, higher education or employment. A key feature of the TechBac® is the Skills Zone, a unique portal that brings together workplace skills training with a mentoring programme, business challenges, and an online CV Builder.

The different systems are linked together using the Experience API (xAPI). Using this approach, City & Guilds are able to store learner's progress in a Learning Record Store (LRS), unifying all of the systems into a single record of progress and achievement.

City & Guilds selected Learning Locker to power their solution, allowing TechBac® to scale upwards of 50,000 learners with peaks of intensive usage throughout the year. Learning Locker's advanced API is used to process and display data to both learners and their tutors, as well as reporting back usage to City & Guilds for analysis.

Learners visualise their work on the City & Guilds Skills Wheel; a personalised dashboard of progress which Learners can export to their own tailored CV, customising the information they present back to potential employers.

Student's progress is monitored through the Performance Optimiser, an interactive dashboard. This system takes data directly from the LRS to provide a real-time benchmark for each cohort undertaking the TechBac®. It allows tutors to highlight areas of risk to the Learner's progress and allows them to intervene at the right time.

Tutors can recognise the Learner's competence in each of the seven skill areas by awarding a Mozilla Open Badge. The Mozilla standard makes accreditation evidence-based and portable. Learners can export their Open Badges to their CV and the wider web as proof of their achievements.

This portability is achieved using the xAPI standard; the first solution to combine both xAPI and Open Badges into a production environment.



VITAL

VISUALISATION
TOOLS AND
ANALYTICS FOR
**ONLINE LANGUAGE
LEARNING (VITAL)**

A collaboration between 3 universities across Europe and HT2 Labs, VITAL is a European Funded initiative founded with the aim of allowing students and teachers to better benchmark language learning across the continent.



Activity Provider

Moodle
Blackboard
Custom Learning Portal



Learning Record Store

Learning Locker (Open Source Edition)



Activity Consumer

Custom Dashboards



Solution

Despite the fact that many HE institutions have embraced eLearning and other learning technology solutions, the Vital consortium has found that there is still no way for either the learning institution or the students themselves to form a clear picture of their online learning habits.

VITAL aims to identify how students of higher education across Europe learn languages online by taking a bottom-up approach learning analytics approach: Mapping and feeding back to the students and their lecturers the existing learning patterns in 4 different types of online language learning.

Easy-to-use dashboards for non-specialist users are a key part of this project to ensure both students and their tutors:

For students so they can not only understand their own data about how they learn online, but also to be able to compare their profiles to those of their peers

For educators to get dynamic and real-time overviews of how their students are progressing, who might be at risk of dropping out or of failing the course, and which parts of the courses cause difficulties/require more feedback.



INTERCONTINENTAL HOTELS GROUP: MEASURING THE VALUE OF **SOCIAL** **LEARNING WITH xAPI**

InterContinental Hotels Group (IHG) commissioned a new Massive Open Online Course (MOOC) to help its first level leaders to have better quality feedback conversations with their reports. The course would run over a 5-week period and encourage learners to contribute back to the online classroom, as well as consuming learning content. The aim was to actually change the way in which leaders gave and received feedback in the workplace, on a global scale.



Activity Provider

Curatr



Learning Record Store

Learning Locker (Enterprise Edition)



Activity Consumer

Custom Dashboards



Solution

IHG wanted to try and measure the actual impact on behaviour of running a MOOC. Without evidence as to the quality of the learning intervention it would be hard to assess the actual impact the MOOC had on the organisation, beyond a range of 'feel good' engagement metrics. IHG felt the MOOC had been successful, but they lacked really hard evidence that it had made a difference to the organisation.

Whilst the MOOC was going on, HT2 Labs was collecting evidence across systems using Learning Locker and the xAPI. With this source of data HT2 Labs' data scientists were able to do something really clever: They were able to analyse conversations retrospectively to see how many people were actually doing something different in the workplace by the end of the 5-week experience. The results were ground-breaking.

Using an approach to measuring the quality of online conversation in a learning environment known as Cognitive Presence, HT2 Labs were able to map the submissions at the 'end of week assignment' to assess the quality of the reflective conversations occurring on a 3-point scale:

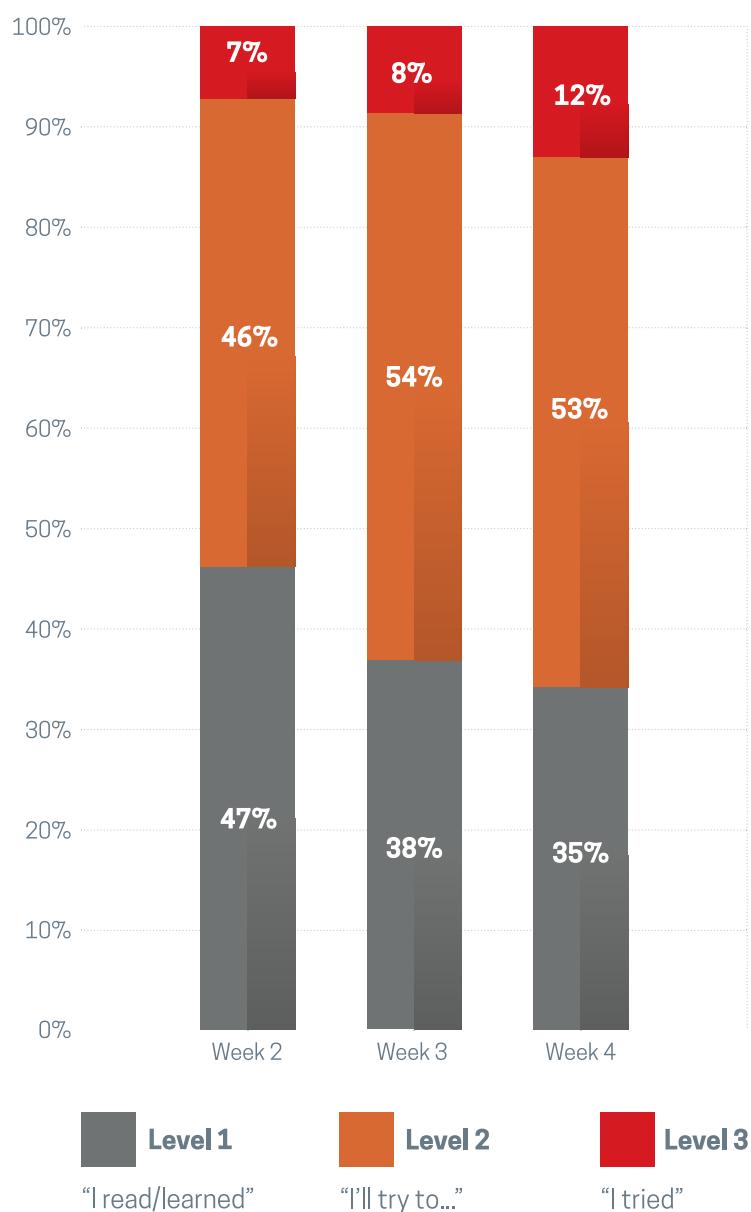
At level 1, learners were simply expressing what they had watched, read or understood during the weeks learning activities

At level 2 they were expressing a desire to change; that they might try to apply some of their new insights at some future point

And at level 3, they were reporting back on how they tried to do something new, reflecting on how it went and on how they might do things next time as they try to form a new habit of better conversations.

The results of this analysis are shown on the following graph:

Within 4 weeks of the social experience starting, IHG could show that 50% of participants had seen how they could change and 12% of participants had actually tried a new behaviour. A remarkable insight into the impact a social learning experience has had on individuals, and collectively, the organisation.



SUMMARY

xAPI's rise to prominence in its first 4 years of existence has never really been seen before in Learning Technology. It took SCORM a decade or more to catch-on and many other specifications have tried (and most often failed) to capture the imagination of the Learning Technology industry. Whilst xAPI might remain firmly at the start of the 'hype cycle' for some, those of us in the know recognise the sea-change that is upon us. Learning Technology and training in general has always been under pressure to prove its value to the organisation. We have no excuses not to measure everything that we do and xAPI gives us a method to start doing just that.

xAPI won't help you to prove the value of something that has no value, quite the opposite in fact. But it will help you showcase the difference training can make to individuals and your organisation as a whole. For most of us, proving such things will be a tricky road. But immediate value can be had from connecting systems together - creating that single source of record that will form your basis for analysis in years to come. And we cannot emphasise enough the importance of starting today; the quality of future analysis will be predicated on the quality of the data you are collecting today. Standardising your approach to this systematic collection of data should be top of your agenda as a Learning Technology Manager. And with this guide at your side, you should be in a good place to get started.

For more resources, including our free MOOC - An Introduction to the xAPI - which has helped thousands of Learning Technology Managers like yourself get to grips with xAPI for the first time, check out the resources section of our website, HT2labs.com/resources.

HT2 Labs is dedicated to helping organisations like yours improve lives through learning technology. We're experts in xAPI and in learning analytics. We developed the world's most installed Learning Record Store, Learning Locker, and we've been the vendor at the heart of some of the most complex xAPI initiatives in the world. We are here to help, no matter how small or large your problem.

To talk with an advisor about how to get started with xAPI, or to kick up a free trial of Learning Locker, please do get in touch today by emailing hello@ht2labs.com



Learning Locker
HT2Labs

LEARNING RECORD STORE



Connect systems together, prove the impact of training and make better decisions on your future learning design.



Learning Analytics



Business Rules



Identity Management



Cloud Deployment



SSL Secured



Encrypted at Rest

Find out more at HT2Labs.com/learning-locker



HT2Labs