zoomi
CAPTURE ANALYZE EVOLVE

ZOOMI INTEGRATION GUIDE

TABLE OF CONTENTS

GOALS

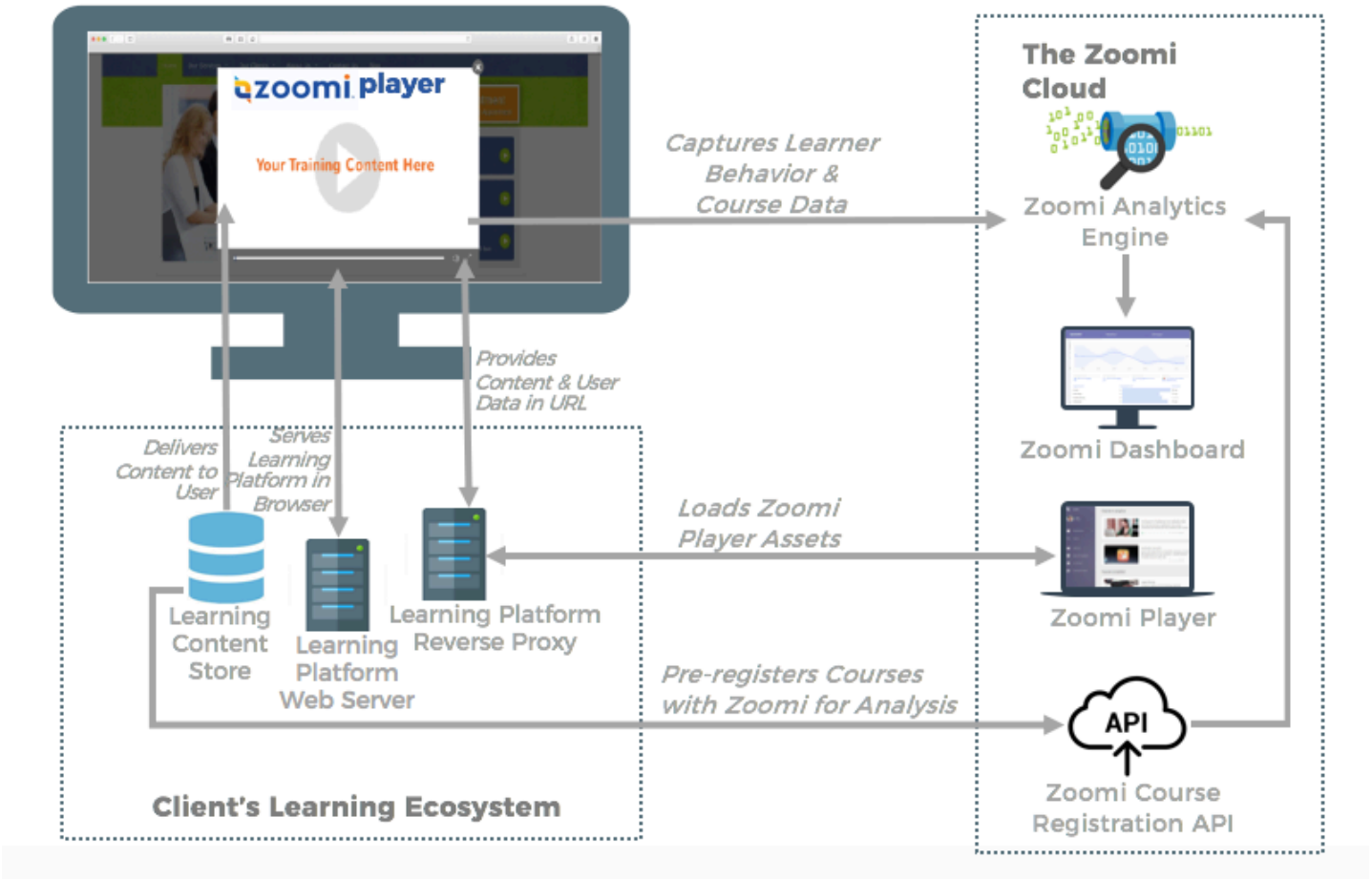
This document is intended to provide an understanding of how the Zoomi Analytics Platform can be integrated with many different types of learning platforms. The learning platform could be your LMS, your learning experience platform, or even a custom-built learning portal that serves e-learning content; in all cases, the integration approach described in this document allows your learning platform to leverage Zoomi's content analysis and learning analytics. Zoomi's goal is to minimize the effort on the part of your development teams while still delivering the full benefits of the Zoomi platform.

The path to integration considers that both your learning platform and the Zoomi platform are cloud based (public or private), and therefore that the primary avenue to integration is via standard web-based (https) APIs. All authentication, information exchanges, and other interactions should be based on accepted security standards. It should also be noted that the Zoomi platform does not need access to any services in your environment.

Because a learning platform and/or learning ecosystem can use a myriad of technologies and can be installed on different operating systems and web servers, the following pages cannot go into specific OS, web server, and learning platform detail; instead, the objective is to expose the necessary integration points. In many cases there are several solutions that will work equally well, for example, authenticating server to server communications. In such cases, this document can be viewed as a first step towards defining the integration approach most suitable to your organization.

ZOOMI PLATFORM INTEGRATION OVERVIEW

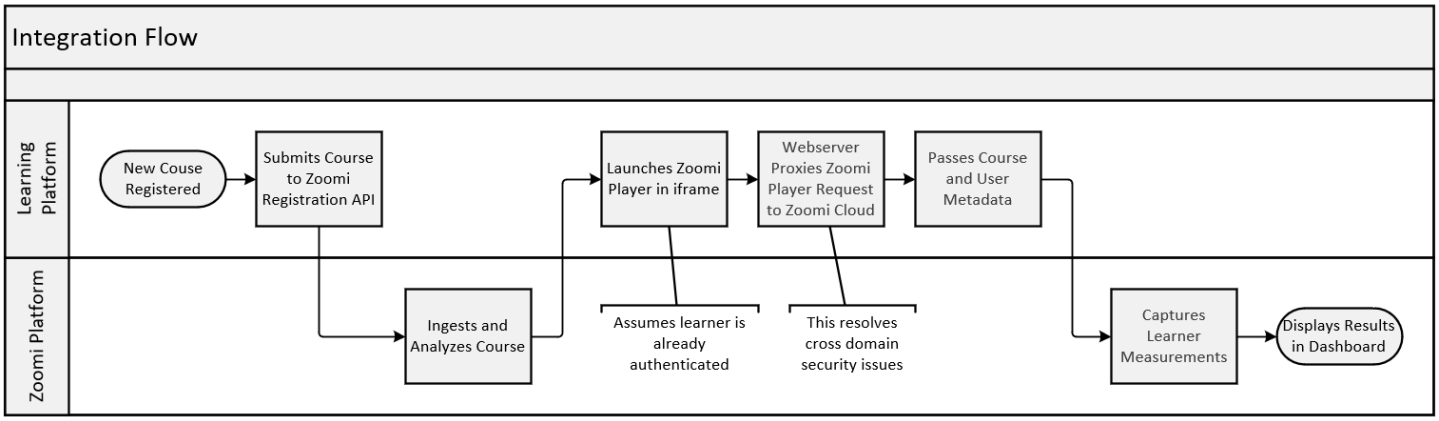
This section provides a high-level overview of the integration. Technical details will follow in additional sections below.



The Zoomi platform captures learner behaviors as learners interact with content in the learning platform. Zoomi then performs analysis on the collected data in order to produce actionable insights at each course level: the entire course, units, content, and segment. A small sample of the analytics produced are: topic analysis, engagement, completion, time spent vs. estimated time, activity, learning paths and views.

To enable the Zoomi Player to collect learner measurements, the Zoomi platform first needs to analyze the course. During this this course registration process, Zoomi applies numerous patented algorithms to course content that deconstruct a course into discrete elements and perform topic analysis. This is typically done by submitting the content to our Course Registration API as part of your course publishing workflow. See the Course Registration section below for more detail.

Once this registration process is complete, the learning platform browser app then loads the Zoomi player within an iframe in order to capture learner measurements. To achieve this, the learning platform launches the Zoomi Player with a URL that contains the metadata needed to access the course. The Player then captures and sends measurements to the Zoomi Analytics Engine, and the results are made available for viewing in the Zoomi Dashboard.



DETAILED INTEGRATION GUIDE

AUTHENTICATION

The Zoomi platform authenticates a company via RFC 7519, the JSON Web Token (JWT) protocol. The JWT is signed and passed to the Player when launched, and Zoomi validates the company by using a public key. JWT is a widely accepted standard, for example used in Google's OpenID Connect service for authentication. Zoomi is using the RS256 standard.

To sign a JWT, it is necessary for each client to generate their own private and public RS256 keys, and share the public key with Zoomi. The client will retain the private key and uses it to sign their API request, while Zoomi retains only the public key, and uses that to authorize the client.

Example key generation steps:

```
ssh-keygen -t rsa -b 2048 -f jwtRS256.key

# Don't add passphrase

openssl rsa -in jwtRS256.key -pubout -outform PEM -out jwtRS256.pub
```

COURSE REGISTRATION

COURSE REGISTRATION OVERVIEW

Zoomi requires the following metadata for proper course registration in our platform. Typically, the course is first registered with the learning platform, and has its own course and content identifiers, including SCORM IDs as applicable. The course files are then uploaded to Zoomi via API along with the following information.

- Course identifier: A unique identifier that will be used to associate a course within the Zoomi system with the course in the Learning platform. The same ID will be used to launch the course.
- Content identifier: A unique identifier that will be used to associate each discrete section of content (if more than one in a course) within the Zoomi system with the content in the Learning platform.
- Course title and version numbers (if there can be multiple versions).
- Content titles and version numbers (if there can be multiple versions).

COURSE REGISTRATION API

The Zoomi Course Registration API is designed to allow learning platforms to submit content, initiate processing and monitor status. Processing course content will allow Zoomi to report detailed analytics on user interaction with course content.

In order to submit a course for analysis and registration, the following steps must be taken:

1. Submit the course metadata to the 'courses' API endpoint.
2. Upload the content for processing by the Zoomi system by posting to the 'contents' API endpoint

POSTING TO ZOOMI API ENDPOINTS

Zoomi Course Registration API endpoints require a valid JSON Web Token embedded in the header of the request to authenticate that the request is from a trusted source. A JSON Web Token is a set of data that has been signed using private and public RS256 keys provided by your company for this purpose. More information about JSON Web Tokens can be found here: https://jwt.io/

Zoomi only requires basic time and a unique uuid as part of the token payload to validate the token is recent and is correctly signed. The required token data consists of a **jti** field which is a generated random UUID and an **iat** field which is an integer Unix epoch value within the last 60 minutes.

Example token data:

```
{
  "jti": "c99832d5-3d8f-4992-81a2-4d3b8921e568",
  "iat": 1516239022
}
```

A JSON Web Token generated using this data will have three parts, a header, a payload and a signature.

Example JSON Web Token:

eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIxMjM0NTY3ODkwIiwiaWF0IjoxNTE2MjM5MDIyfQ.b9wXpKSAwUgvz7xcwcQF9qaovUINjXPJF78c9CyTYUozMWu4-c2cJLyZT7eqCY4LZgmOYJo2SNNVpDb53F2b1aSK2jrVhO8WBst2XzlE011K56-brJj4QaaSo9STCQPj_o82N-b2XD0tCENjTAWwTaTYXqHtybW9gJza52WrZmE

This token should be embedded in the request header's authorization property, like so:

Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIxMjM0NTY3ODkwIiwiaWF0IjoxNTE2MjM5MDIyfQ.b9wXpKSAwUgvz7xcwcQF9qaovUINjXPJF78c9CyTYUozMWu4-c2cJLyZT7eqCY4LZgmOYJo2SNNVpDb53F2b1aSK2jrVhO8WBst2XzlE011K56-brJj4QaaSo9STCQPj_o82N-b2XD0tCENjTAWwTaTYXqHtybW9gJza52WrZmE

Additionally, an "Accept" header must be included. Only "application/json" is acceptable at this time.

Example request header:

Accept: application/json

Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIxMjM0NTY3ODkwIiwiaWF0IjoxNTE2MjM5MDIyfQ.b9wXpKSAwUgvz7xcwcQF9qaovUINjXPJF78c9CyTYUozMWu4-c2cJLyZT7eqCY4LZgmOYJo2SNNVpDb53F2b1aSK2jrVhO8WBst2XzlE011K56-brJj4QaaSo9STCQPj_o82N-b2XD0tCENjTAWwTaTYXqHtybW9gJza52WrZmE

POSTING COURSE METADATA

The first step in submitting a course for processing by Zoomi is POSTing the course metadata to the 'courses' API endpoint. This data contains the basic structure and information about the various tiers of your course. Zoomi's course structure consists of several 'tier' levels: course, module, unit, and content. Each of the tiers below course can contain multiple child items. So a course could have multiple modules, a module can contain multiple units, and a unit can contain multiple contents. Zoomi only requires basic information about each item in a tier to inform on course structure: id and title, but additional fields can provide Zoomi with more information about your course. See below for attribute definitions. Once course metadata has been submitted course data with the same course id cannot be submitted again without cancelling the course import job. How to cancel is discussed later in this document.

The course JSON must be posted to Zoomi's course metadata endpoint using a standard HTTP POST request. The request must contain the authorization header as detailed above and the course metadata JSON as described here. The request should be POSTed to the 'courses' API endpoint located here:

https://management.zoomiinc.com/api/v4/courses

Example course JSON request payload:

```
{
        "id": "some_course_id",
        "title": "Some course title",
        "group": "some_group",
        "description": "This course...",
        "supplementary_docs": {
                "label": "sup_doc_label",
                "items": [
                        {
                                "id": "sup_doc1"
                        }
                ]
```

```
        },
        "modules": [
                {
                        "id": "module_1_id",
                "title": "module 1 title",
                "units": [
                                {
                                        "id": "module_1_unit_1_id",
                        "title": "some unit title",
                        "contents": [
                                                {
                                                        "id": "module_1_unit_1_content_1_id",
                                "title": "some content title"
                        }
                                        ]
                                }
                        ]
                }
        ]
}
```

A request containing the above data would create a course with a single module, unit, and content item (and an additional supplementary document) using the ids and titles to inform on those items. A sample request using this data may look like:

```
curl -H "Accept: application/json" -H "Authorization: Bearer
eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJqdGkiOiIxMjM0NTY3ODkwIiwiaWF0IjoxNTE2MjM5MDIyfQ.b9wXpKSAwUgvz7
xcwcQF9qaovUINjXPJF78c9CyTYUozMWu4-c2cJLyZT7eqCY4LZgmOYJo2SNNVpDb53F2b1aSK2jrVhO8WBst2XzlE011K56-
brJj4QaaSo9STCQPj_o82N-b2XD0tCENjTAWwTaTYXqHtybW9gJza52WrZmE" -X POST
"http://management.zoomi.com/api/v4/courses" -d '{"id":"some_course_id","title":"Some course
title","group":"some_group","description":"This
course...","supplementary_docs":{"label":"sup_doc_label","items":[{"id":"sup_doc1"}]},"modules":[{"id":"module_1_id",
"title":"module 1 title","units":[{"id":"module_1_unit_1_id","title":"some unit
title","contents":[{"id":"module_1_unit_1_content_1_id","title":"some content title"}]}]}]}'
```

Attribute definitions for course metadata:

Attribute	Definition
id	A unique identifier value string for the course. Maximum length 60 characters. This identifier must only contain numbers, letters, and underscores. It is required that this identifier be unique to Zoomi's system, so we recommend prefixing the identifier with some string unique to your company. (example: "YOURCOMPANY_COURSE_1")
title	This is a title string to describe your course
group (optional)	A group identifier string that can be used to group courses together for analytics purpose. This is useful if you are submitting multiple courses that should be associated, different versions of the same content, for example.
description (optional)	A short description of your course.
supplementary_docs (optional)	A JSON object containing a string label and an array of one or more objects, each containing a supplementary document ID. The label applies to all of these documents, whose IDs are purely to differentiate them when uploading.
modules (optional*)	An array of modules for your course. *If your course only has a single module, this can be omitted, and a "units" property can be used in its place.

Attribute definitions for module metadata:

Attribute	Definition
id	A unique identifier value string for each module. Maximum length 60 characters. This identifier must only contain numbers, letters, and underscores. It is required that this identifier be unique to Zoomi's system, so we recommend prefixing the identifier with the course identifier of the parent course. (example: "YOURCOMPANY_COURSE_1_MODULE_1")
title	This is a title string to describe this module. Typically, modules represent separate topics in a course
units	An array of units for the course. Each unit typically represents a chapter of your course.

Attribute definitions for unit metadata:

Attribute	Definition
id	A unique identifier value string for each unit. Maximum length 60 characters. This identifier must only contain numbers, letters, and underscores. It is required that this identifier be unique to Zoomi's system, so we recommend prefixing the identifier with the module or course identifier of the parent. (example: "YOURCOMPANY_COURSE_1_MODULE_1_UNIT_1" or "YOURCOMPANY_COURSE_1_UNIT_1")
title	A title string to describe this unit.
contents	An array of content items for the unit. Each content item will be a piece of content for display and interaction by your learners.

Attribute definitions for content metadata:

Attribute	Definition
id	A unique identifier value string for each content item. Maximum length 60 characters. This identifier must only contain numbers, letters, and underscores. It is required that this identifier be unique to Zoomi's system, so we recommend prefixing the identifier with the unit identifier of the parent. (example: "YOURCOMPANY_COURSE_1_MODULE_1_UNIT_1_CONTENT_1" or "YOURCOMPANY_COURSE_1_UNIT_1_CONTENT_1")
title	A title string to describe this content item.

COURSE METADATA API RESPONSE

Once a POST to the 'courses' API endpoint has been made a response will be returned indicating a management URI and a list of upload URI(s). The management URI is used for checking on the status of the course import job as it is processed or deleting the import job to cancel the course processing. The upload URI array will contain a list of upload endpoints for each content item specified in your course metadata. The upload URI(s) will allow sending of course content to Zoomi's system for processing. This is detailed in a below section and must be completed as a next step for the course import job to continue.

Sample response data:


```
{
  "status": "success",
  "data": {
    "management_uri": "https://management.zoomiinc.com/api/v4/courses/imports/some_course_id",
    "upload_uris": {
```

"module_1_unit_1_content_1_id": "https://management.zoomiinc.com/api/v4/contents/526ab0bf-cd6b-477e-91e6-698c362b1fd9",

"module_1_unit_1_content_2_id": "https://management.zoomi.com/api/v4/contents/e554afda-dd78-445e-9587-494e2abc22ba"

}

}

}

UPLOADING CONTENT FOR PROCESSING

Once the content metadata has been successfully submitted to the Zoomi API for processing, using the response from that request, content will need to be uploaded to each of the upload URI(s) provided. If all content for a course is not uploaded in a period of 30 days, content must be re-uploaded to the same URI(s) provided in the metadata POST response as detailed above.

In order to POST to each of the provided endpoints each request must be done separately and contain the appropriate headers, as detailed in the first section of this document. Each content upload endpoint will contain a unique identifier for that content (as seen in the urls above).

Example content upload uri:

https://management.zoomiinc.com/api/v4/contents/e554afda-dd78-445e-9587-494e2abc22ba

The contents should simply be the file being uploaded.

Example upload request:

curl -X POST https://management.zoomiinc.com/api/v4/contents/e554afda-dd78-445e-9587-494e2abc22ba -T video_1.mp4

CONTENT UPLOAD API RESPONSE

If successful, your upload will receive a response containing the following JSON data:

{ "status": "success", "data": {} } }

SUPPORTED CONTENT TYPES

The Zoomi course processor supports a limited amount of content types for upload: pdf, video (with or without subtitles), and various types of SCORM 1.2 compliant course content. Please see the following support document for additional information:

https://support.zoomiinc.com/hc/en-us/articles/360007513454-Zoomi-Compatibility

If the content contains multiple files (for example a Captivate SCORM 1.2 course), it must be compressed before being uploaded to the endpoint with the directory structure of the content intact.

REQUESTING A COURSE IMPORT JOB STATUS

At any time in the import process the status of the import job may be requested but making a GET request to the management URI provided in the response from the course metadata request, as detailed above. The GET request must contain valid headers as detailed at the beginning of this document. The GET request to the management URI will provide overall status and status information for each content item.

COURSE STATUS API RESPONSE


```
{
        "phase": "20 : ReadyForIndexing",
        "content_status" [
                "upload_uri": "https://management.zoomi.com/api/v4/contents/526ab0bf-cd6b-477e-91e6-698c362b1fd9",
    "content_id": "module_1_unit_1_content_1_id",
                "uploaded": True,
                "uploading": False
        ]
}
```


The 'phase' has a number of different values and will indicate where the course as a whole is in the importing process.

Phase Definitions

Phase	Definition
Registered	The initial course metadata has been POSTed, the content is waiting to be uploaded
ReadyForExtraction	All of the content has been uploaded to the content upload uris and the course is ready to be extracted and processed
Extracting	Any course zips are being extracted and the course content is being uploaded to the Zoomi file store for processing
ReadyForIndexing	Content has been extracted successfully and is ready to be indexed for use with our analytics functionality
Indexing	Content is being indexed individually by examining the specific types of content and breaking it down for analytics handling
ReadyForScormRegistration	Content has been indexed and SCORM 1.2 content is waiting to be registered with Rustici SCORM engine 2017

ScormRegistering	SCORM 1.2 content is being registered with SCORM engine so that it can be properly launched and SCORM data gathered
ReadyForDbInsertion	All of the data that is needed has been gathered, content files are in place and the necessary DB records are ready to be written
DbInserting	DB records are being written
Complete	Import is complete
CompleteAndNotified	Users have been notified that the import has successfully completed
Error	An error somewhere in the import process has occurred. Course import has been halted
ErrorAndAlerted	User have been notified of the error

CANCELLING A COURSE IMPORT JOB

Before the content has been uploaded, or if the job has errored, a course import job can be cancelled. To do so simply make a DELETE request to the management uri provided in the response from the course metadata request, as detailed above. The DELETE request must contain valid headers as detailed at the beginning of this document.

Example course import cancellation request:

curl -X DELETE https://management.zoomiinc.com/api/v4/courses/imports/some_course_id

This will cancel the job and delete any records created in the Zoomi system as well as all files uploaded.

If successful, your request will receive a response containing the following JSON data:

{ "status": "success", "data": {} } }

PREPARE THE WEB INFRASTRUCTURE

CONFIGURE A WEB PROXY TO LAUNCH THE ZOOMI PLAYER

Learning platform web applications launch courses via the Zoomi Player iframe. To avoid cross-site scripting issues that arise when using SCORM the learning platform web application must launch the Zoomi Player within the learning platform's domain. This is accomplished by configuring a reverse proxy on the Learning platform web server.

For example, if the Learning platform is launched from

https://lms.mycompany.com

The base reverse proxy to the Zoomi Player must listen to an URL like this in the same domain

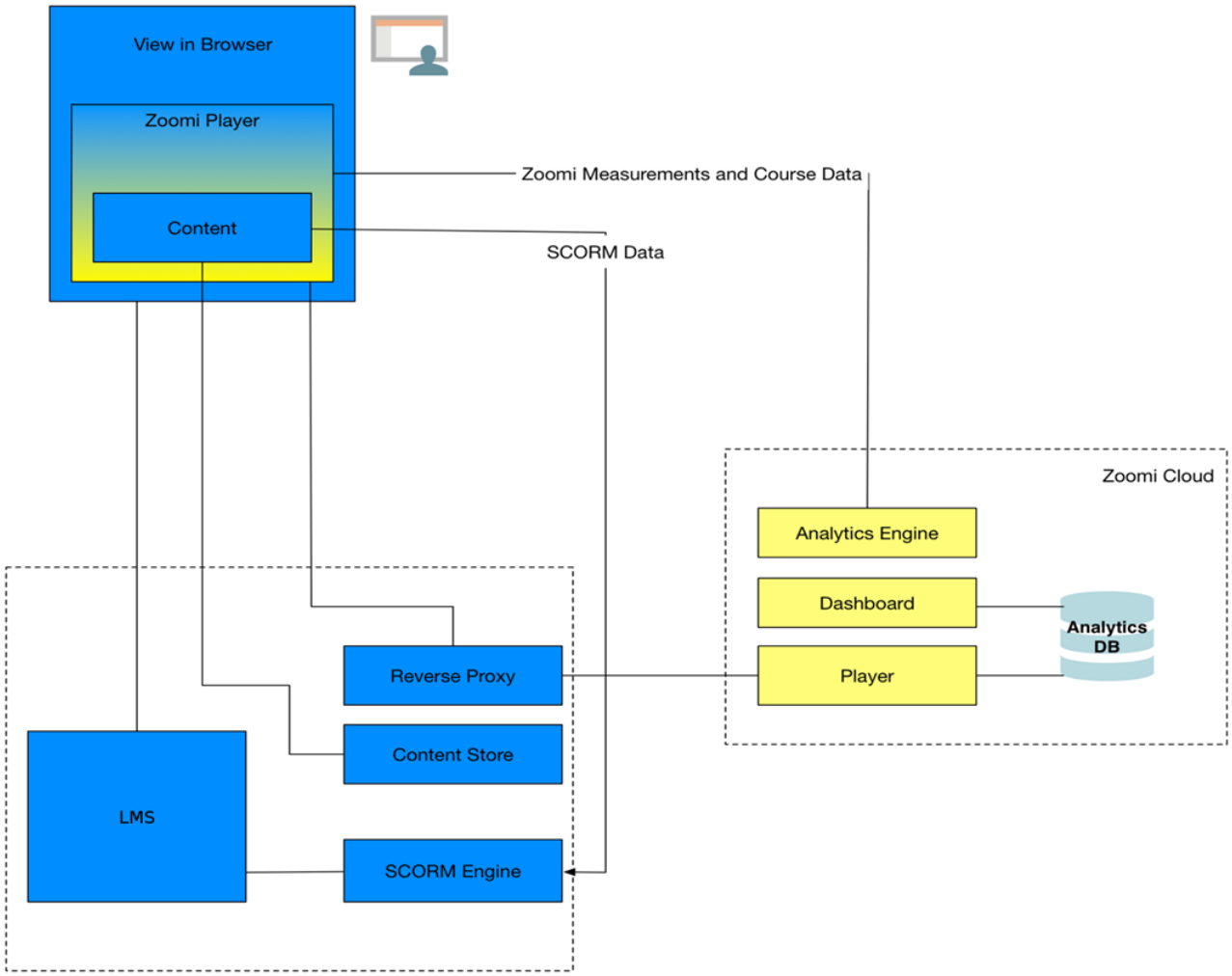
https://lms.mycompany.com/zoomi/?args

The reverse proxy will fetch resources with this base URL from Zoomi's platform via a client's URL, for example:

http://mycompany.zoomiinc.com/?args

The amount of transmitted data (HTML and JavaScript files) is quite small and this setup can easily scale.

It has been successfully implemented with Nginx, Apache and ISS. If required Zoomi can provide further advice.



GENERATE A JSON WEB TOKEN FOR AUTHENTICATION

As described in the Authentication section, JWT must be generated and signed with the client's private key. The Zoomi Player requires user and course information on launch. This information can either be included in the JWT (if using Option 2 or Option 3 below) or passed as part of the launch URL (Option 1).

OPTION 1 – CREATE A STATIC JWT THAT IS PASSED BY THE WEB PROXY

NOTE: This option cannot be used if the learning platform integration is intended to support more than a single company/tenant in which case the Dashboard will show all courses offered by the learning platform and all learners are in a single organization.

A static JWT can be generated which is then always configured into the reverse proxy configuration. The proxy authenticates requests to the Zoomi platform by appending an argument "jwt=[JWT]" to each request's URL.

This JWT will only contain a unique company ID assigned by Zoomi and the "iat" (issued at) timestamp attribute as the payload.

If this method is chosen, the user and course information must be passed as additional URL arguments when the Learning platform Web application launches the Zoomi Player iframe.

OPTION 2 – DYNAMICALLY CREATE A JWT FROM THE WEB APPLICATION

This option can be used to support multiple companies from a single Learning platform. The service that dynamically creates the JWT will transmit the correct company_id along with other attributes.

With this option, a JWT web service is implemented. JavaScript code in the Learning platform web application encapsulates company, user, and course information and then makes an asynchronous call to the JWT web service to obtain a signed JWT which is appended as an argument to the Player's launch URL. An example of a JSON payload looks like this:

```
{"company_id": "38738",
 "user_id": "1000284302",    // any unique user identifier
 "first_name": "John",        // optional
 "last_name": "Smith",       // optional
 "email": "john.smith@lms.mycompany.com",     // optional
 "course_id": "LMS_AB123",
 "iat": 1520706494

}
```

You can view the URL, including the JWT, used to launch the Zoomi Player iframe in the following section. This JWT for Option 1 contains at a minimum a unique company ID assigned by Zoomi, a user_id, a course identifier, and the "iat" timestamp attribute as the payload

OPTION 3 – MODIFY THE LEARNING PLATFORM TO GENERATE LINKS THAT INCLUDE THE JWT

This option can be used to support multiple companies/tennants from a single Learning platform if the JWT is dynamically generated by the Learning platform. The service that dynamically creates the JWT will transmit the correct company_id along with other attributes.

The most straightforward method If you have access to your Learning platform source code is to make a small modification to your Learning platform to dynamically create the JWT (with the same attributes as in Option 2) and include it as part of the Learning platform links generated for the Web application, for example in course listings.

You can view the URL, including the JWT, used to launch the Zoomi Player iframe in the following section.

This JWT will always contain a unique company ID, a user_id, a course identifier, and the "iat" timestamp attribute as the payload (the same as in Option 2).

RECOMMENDED JSON WEB TOKEN LIBRARIES

Language/framework	Link
.NET	https://github.com/AzureAD/azure-activedirectory-identitymodel-extensions-for-dotnet
Python	https://github.com/jpadilla/pyjwt/
Java	https://github.com/auth0/java-jwt

LAUNCH THE COURSE IN THE ZOOMI PLAYER

Instead of the Learning platform web application directly launching the SCORM content, the Zoomi Player is launched as either an embedded iframe or a new window. As described in the sections above, the URL used to launch the Player must contain user identification and course information, depending on which option was chosen in the previous section. The method for passing this information will depend on the option chosen for creating the JSON Web Token. Note that the "cl" argument in the example URLs is the SCORM launch link.

- If Option 1 was chosen (a static JWT appended by the reverse proxy) then user and course attributes must be formatted as arguments in the Zoomi Player launch URL.

https://lms.mycompany.com/zoomi/?user=1000284302&course=LMS_AB123&cl=https%3A%2F%2Flms.mycompany.com%2FScormEngineInterface%2Fdefaultui%2Flaunch.jsp%3Fregistration%3Dc669ee29-22f2-444a-8493-879df1f91ad8

- If Options 2 or 3 were chosen, then the user and course information will be encapsulated in the JWT which will be a Base64 string that is appended as an argument to the GET request and as shown in the JSON example in the previous section.

https://lms.mycompany.com/zoomi/?jwt=ew0KICAidXNlcl9pZCI6ICIxMDAwMjg0MzAyIiwNCiAgImZpcnN0X25hbWUiOiAiSm9obiIsDQogICJsYXN0X25hbWUiOiAiU21pdGgiLA0KICAiZW1haWwiOiAiam9obi5zbWl0aEBsbXMubXljb21wYW55LmNvbSIsDQogICJjb3Vyc2VfaWQiOiAiTE1TX0FCMTIzIiwNCiAgImlhdCI6IDE1MjA3MDY0OTQgIA0KfQ0K&cl=https%3A%2F%2Flms.mycompany.com%2FScormEngineInterface%2Fdefaultui%2Flaunch.jsp%3Fregistration%3Dc669ee29-22f2-444a-8493-879df1f91ad8

These are the possible launch URLs for Zoomi's Player. The reverse proxy reacts on the /zoomi/ path component and rewrite the URL by replacing the domain and the /zoomi/ component with Zoomi's domain and requests the resources from there.

LEARNER REGISTRATION ON THE ZOOMI PLATFORM

The Zoomi platform auto-registers learners:

1. The learner first authenticates with the learning platform

2. The learning platform Web application launches the Zoomi Player with a unique user string using one of the methods already described.
3. If the Zoomi platform does not already contain the user, they are automatically registered.

The user string that identifies the learner is an alphanumeric string that is used to uniquely identify the learner. It is not necessary to pass identifiable user information such as email address or first and last name if you wish to refrain from transferring personally identifiable user information to the Zoomi platform, as long as the identifier is:

- Unique to the user
- Consistent for the user across sessions and across courses

For example, an employee number can be used as well as any other unique valid string. It's important to remember, however, that if user names are not supplied you will be unable to identify individual users in the Zoomi Analytics Dashboard or search for names in reports.