

OEA Technical Design

Published: April, 2023

This document provides a detailed explanation of the OEA framework and component architecture along with links to references for detailed info and training on topics such as lakehouse architecture, spark, delta lake, and synapse.

Intro.....	1
1) Setting the context	1
2) Technical Goals of OEA.....	2
3) OEA lakehouse reference architecture.....	2
4) OEA framework	4
4.1 The Infrastructure.....	4
4.2 The OEA Framework assets.....	5
5) OEA deployable components	7
5.1 Modules.....	7
5.2 Schemas.....	7
5.3 Packages.....	7

Intro

The **Open Education Analytics** framework provides a modern data lakehouse framework and set of modular components that greatly simplify the process of setting up and leveraging a modern data estate to deliver nudges, dashboards, reports, and AI models that provide insights from your data. The OEA framework as well as the assets built on the framework are all open-source under an MIT license. The ongoing buildout of these assets is being done by partners and customers in collaboration with Microsoft – on a global scale. The goal here is for the global education ecosystem to have a clear path for modernizing data estates in education, and to have a clear way of sharing and building on the work of others in the community.

1) Setting the context

We face numerous challenges when it comes to data centralization, data processing, data interoperability, and data analytics. We can break down the most direct/immediate challenges into 3 broad areas:

- 1) extraction – pulling data from a large number of siloed systems, with different interfaces, on a recurring basis
- 2) processing – applying compute to extracted data to make it usable and add value (validate, cleanse, explore, aggregate, enrich, map, analyze)
- 3) consumption – serving the data to support consuming systems and visualizations

Added to these challenges are the cross-cutting concerns of:

- 4) process orchestration
- 5) resource management and monitoring
- 6) security and role-based access control
- 7) data governance
- 8) operationalizing of technical assets (version control, deployment process)

With the advent of technologies including Spark and Delta Lake, and cloud services like Azure Synapse and Azure DevOps and Github – how these challenges are addressed today is far ahead of how it was done with a data warehouse and ad-hoc transformation scripts. It is now possible to address these challenges and work with all types of data (big data, small data, structured, unstructured, semi-structured, batch, real-time) within a single “lakehouse” architecture – and leverage the flexibility and low cost of a data lake while still leveraging the power and usability of a data warehouse. However, as powerful as this newer set of technologies is, the setup and maintenance requires a significant level of effort, made more difficult by the number of technical setup decisions that must be made.

2) Technical Goals of OEA

The challenges in the previous section speak to the need to have a common approach that is well defined and incorporates best-practices and ideally allows for sharing of common components within the global education community. We need the easy button, and we need an ever-growing catalog of reusable components.

OEA is the result of an ongoing effort to facilitate collaboration within the global edtech ecosystem to develop the easy button to address the challenges listed in the previous section and to harvest and curate a catalog of reusable components.

So then there are 3 ways to greatly simplify what is needed to implement and use a lakehouse within the education sector:

- 1) define a lakehouse reference architecture
- 2) build a common lakehouse framework (to allow for interchangeable components within the framework)
- 3) curate catalogs of reusable assets (extraction pipelines, transformation scripts, orchestration pipelines, dashboards, etc)

These are the 3 areas that OEA is developing in collaboration with partners and customers around the world – resulting in open-source assets published directly to the [OEA Github repo](#).

It's important to note that:

- OEA is not a Microsoft product – it is a community facilitated by Microsoft and a set of open-source assets (created by the community and Microsoft) managed and curated by Microsoft
- OEA is schema-agnostic. We do not advocate a specific schema as a core requirement; instead we provide a catalog of schemas to choose from and encourage the community to contribute back additional schemas.
- OEA supports and encourages the use of standards (standard API's, standard edu schemas), but we recognize the need to go beyond what the various edtech standards currently provide. We hope to be a catalyst for the acceleration and convergence of edu standards, globally. Standards are important, but we need to deliver value today while we derive the standards that will make everything easier in the future.

3) OEA lakehouse reference architecture

There are many ways to setup a lakehouse, and there are many best-practices that help you steer clear of common challenges. The OEA lakehouse reference architecture defines the data lake structure and the recommended data processing strategy. This reference architecture also relies on specific technologies (non-proprietary, all open-source) which serve as fundamental cornerstones of a lakehouse architecture – namely [Spark](#) and [Delta Lake](#).

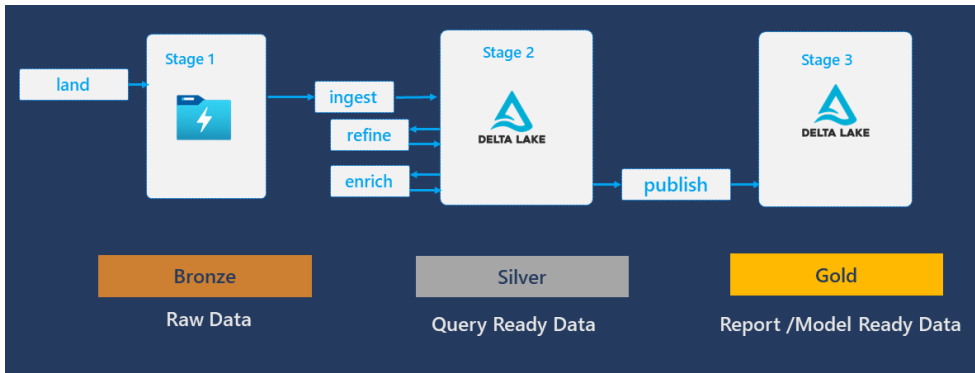
At the highest level, the data lake is divided into three stages, referred to by sequential number or using the terms bronze, silver, gold.

Stage 1 (bronze) is used for landing raw data – in the format it was in when extracted from the source system. Keeping a history of the raw extracts allows for reprocessing of the data when/if necessary. We refer to the act of extracting data and storing it in stage 1 of the data lake as “landing” the data.

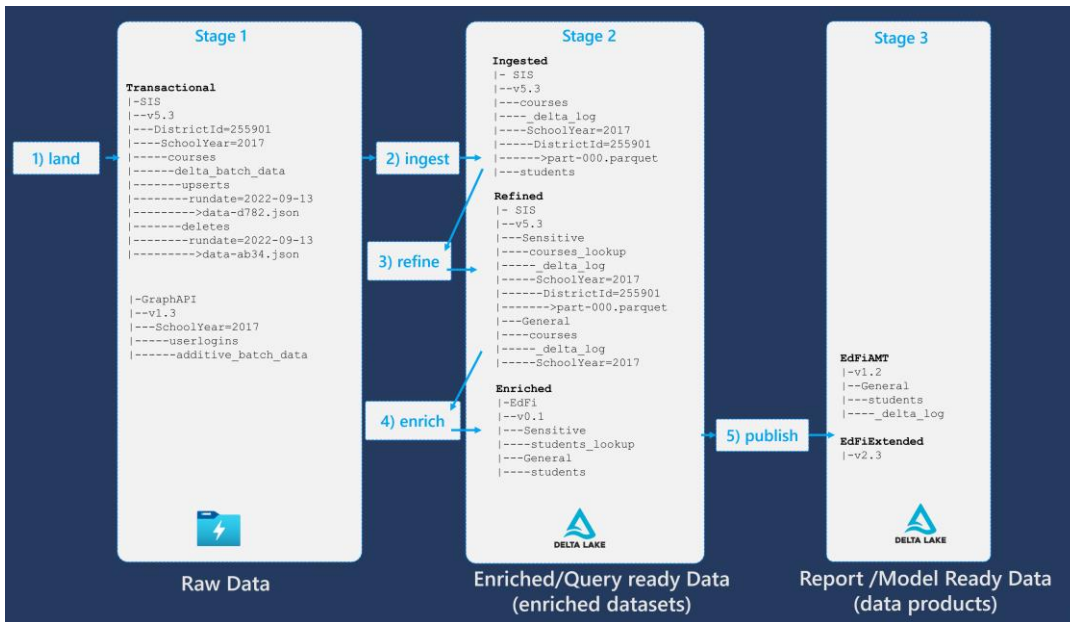
Stage 2 (silver) is where all of the data processing is done. The initial loading of data into stage 2 is referred to as “ingesting” – and is limited to accomplishing 2 key tasks: 1) parse the latest batch of raw data in stage 1, and 2) merge it into the delta lake table for that data source.

Once ingested, the data can be refined and iteratively enriched to produce other delta lake tables and/or map to a standard.

Stage 3 (gold) is the final stage that is used for finalized “data products” – which are relied upon by data consumers (reports, dashboards, ML models, down stream systems, etc). The term used for processes that write to stage 3 is “publish” – denoting a process of optimizing the data structure for consumption (eg, calculating aggregations, mapping to a star schema).



The recommended folder structure within each stage is shown in the diagram below, and is designed to align with the guidance from: ["Data lake zones and containers"](#).



4) OEA framework

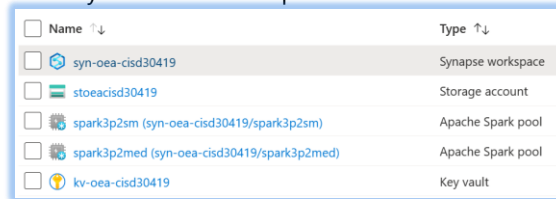
The OEA framework aligns with the OEA reference architecture and provides a set of assets that simplifies the process of setting up and working with a modern data estate built around a lakehouse architecture on Synapse. So in a sense the OEA framework can be seen as an Azure implementation of the OEA reference architecture, but there are many aspects of the implementation that are cloud-agnostic.

In this section we will review the technical details of the framework. For info on setting up and using the OEA framework see the OEA User Guide.

4.1 The Infrastructure

The OEA framework utilizes Azure resources provisioned in your Azure subscription. The minimum set of Azure resources are:

- 1) Synapse workspace
- 2) data lake storage
- 3) spark pool
- 4) key vault



Name	Type
syn-oea-cisd30419	Synapse workspace
stoeacisd30419	Storage account
spark3p2sm (syn-oea-cisd30419/spark3p2sm)	Apache Spark pool
spark3p2med (syn-oea-cisd30419/spark3p2med)	Apache Spark pool
kv-oea-cisd30419	Key vault

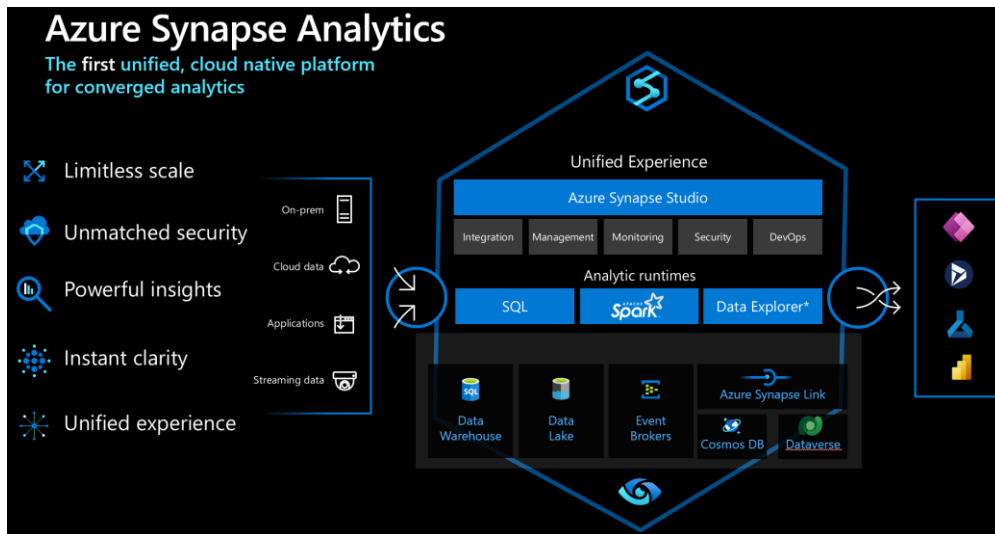
The setup script for OEA will provision these resources for you within a single resource group and will then install synapse assets (pipelines, scripts, integration datasets) directly into the newly created synapse workspace. For details on how the Azure services are provisioned and configured, refer to the [setup_base_architecture.sh](#) script. Note that you can also opt to use an existing synapse workspace – in which case you can skip the provisioning of azure resources and install the synapse assets into your existing synapse workspace ([see Setup of framework assets for info on this option](#)).

These Azure resources are all serverless in nature (there's no dedicated compute being provisioned here), with no licensing costs. The cost incurred is based on usage, meaning that you can setup the OEA framework and begin working with it a minimal cost. For more info on costs see the [OEA cost estimation sheets](#).

Additional Azure resources can be added to further develop the capabilities of your modern data estate – including Azure Purview, Azure Event Hubs, Azure ML, and Azure OpenAI. As OEA continues to develop we will incorporate more guidance and utilities to simplify the leveraging of these and other Azure resources – based on the work being done with customers and partners. For more info on including these additional Azure resources see: [Analytics end-to-end with Azure Synapse](#).

Note that the primary focus of the OEA framework is in developing the pipelines and scripts that are used within synapse to simplify data engineering and data science – not in the setup and maintenance of the underlying infrastructure. The OEA framework provides a setup script that includes the provisioning of the necessary azure infrastructure as a means for getting started quickly, but providing detailed guidance on enterprise scale infrastructure or comprehensive cloud-adoption strategies is beyond the scope of this work. For info on these topics see: [Cloud-scale analytics](#) and [Cloud Adoption Framework](#).

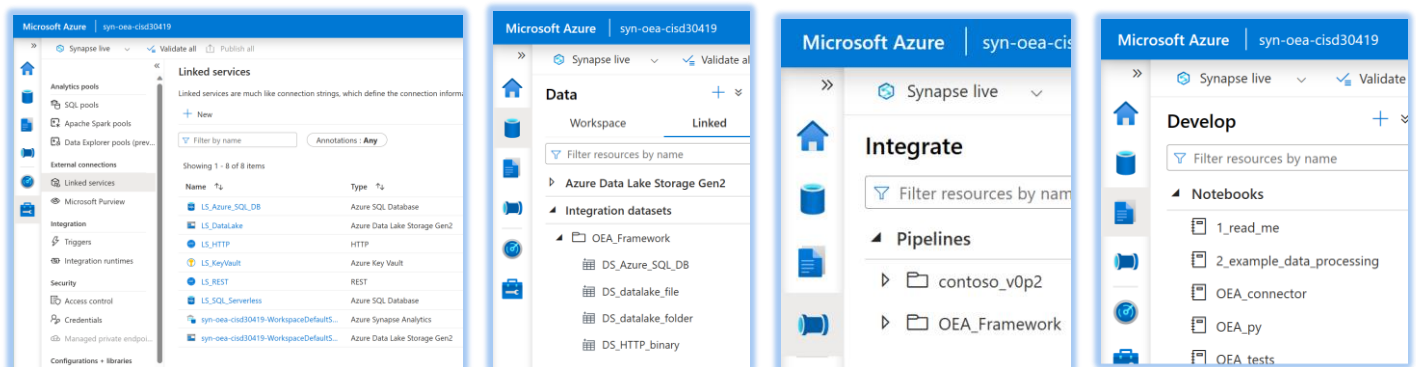
That being said, it's important to note that the necessary infrastructure is being made less and less complex over time. OEA is being positioned to leverage these advances as they roll out and stay aligned with the Synapse approach to data engineering and data science as depicted in the diagram below:



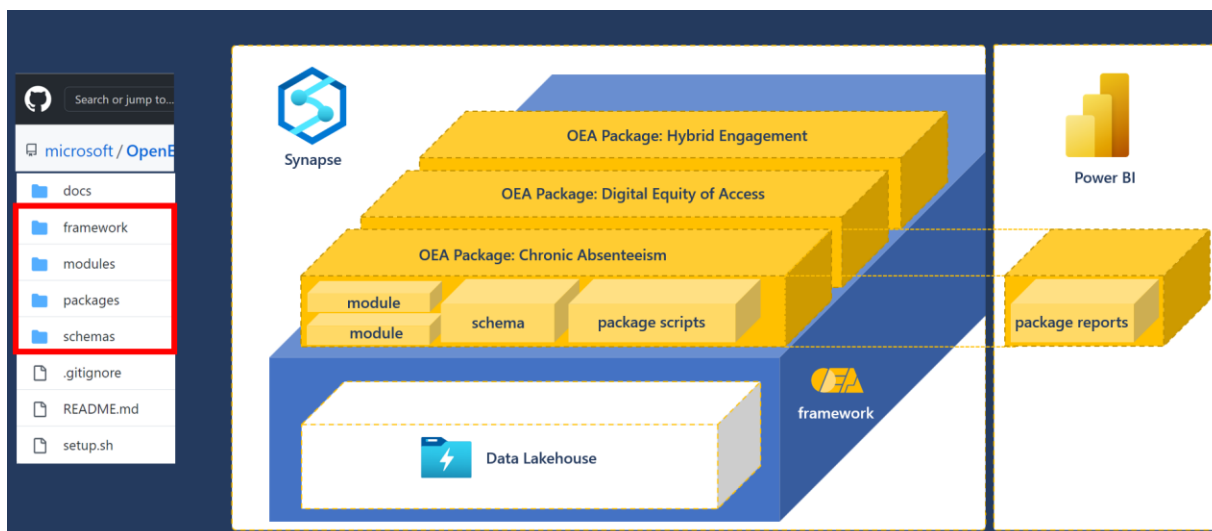
4.2 The OEA Framework assets

The OEA Framework assets are the common assets deployed to the synapse workspace that serve as a set of core utilities that simplify the data engineering process within a lakehouse architecture (the assets can be found at [OpenEduAnalytics/framework/synapse](https://openeduanalytics.com/framework/synapse)).

The core assets include Linked Services, integration datasets, pipelines, and notebooks.

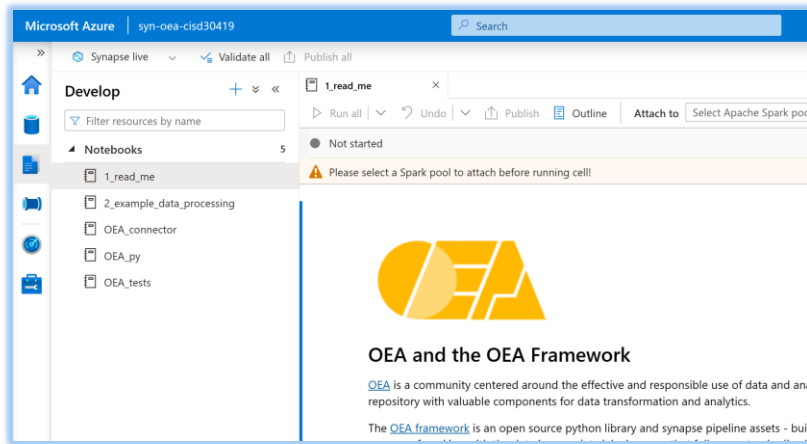


This set of framework assets allow for the creation of modular components that follow the design standards of the framework. The result is a framework that facilitates the building and reuse of deployable components within synapse.



As shown in the diagram above, the Github repository is structured to align with the architecture such that the framework is distinct from the deployable components, and therefore has its own release cycle. The deployable components are comprised of modules, schemas, and packages and will be discussed in the next section.

The best way to learn how to leverage the core framework components is by walking through the included examples as explained in the included notebook named "1_read_me".



5) OEA deployable components

The OEA framework makes it possible to reuse common components that have been contributed by the community. The 2 fundamental components are:

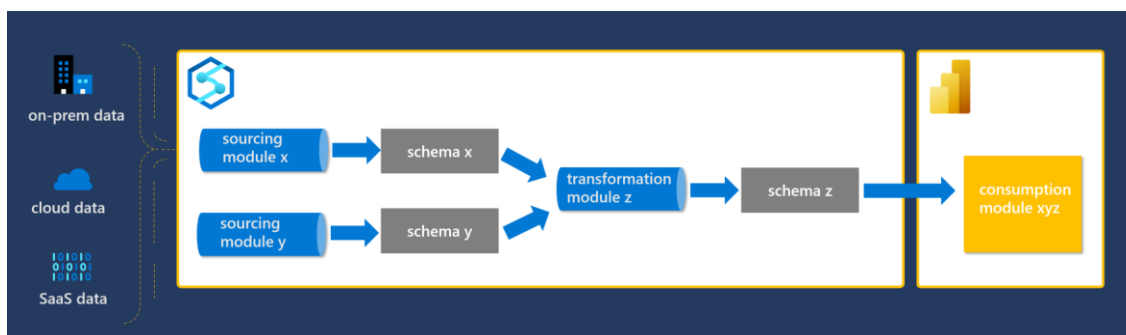
1. modules – consisting of assets that source data, process data, and/or consume data (eg, reports)
2. schemas – a definition of the data at rest

The final component type is simply a bundling of these 2 fundamental types, for the purpose of creating a deployable solution that combines multiple components – this is referred to as a package.

Modules can be conceptualized as processes, schemas as states, and packages as bundled solutions.

5.1 Modules

Modules contain a set of deployable assets that work with data. There are 3 different types of modules: sourcing, transformation, and consumption.



1. **sourcing module** – extracts data from a source system and ingests the data into the lakehouse. Sourcing modules define one or more outputs in the form of a schema.
2. **transformation module** – processes data (validate, cleanse, aggregate, enrich) and stores the results in the lakehouse. Transformation modules declare input dependencies and generated outputs (both defined as schemas).
3. **consumption modules** – utilizes data (eg, dashboards, reports, ML models, outbound pipelines the push data to other systems)

5.2 Schemas

Schemas define data at rest – as a logical grouping of entities and their attributes.

5.3 Packages

Packages are a bundling of components that together provide a solution for a given set of use cases.

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein. The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication. This white paper is for informational purposes only. Microsoft makes no warranties, express or implied, in this document. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation. Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2023 Microsoft Corporation. All rights reserved