

2. Technische architectuur



Deze technische architectuur is beschreven in het blog: <https://blog.surf.nl/learning-analytics-experiment-architectuur-standaarden/>

De technische architectuur (zie onderstaand figuur) van het learning analytics experiment is onder te verdelen in vier lagen:

- De presentatie-laag die de visualisatie verzorgd.

De visualisaties zijn het resultaat van het learning analytics experiment, die docenten en studenten inzicht geven in het studiegedrag. Deze visualisaties worden bijvoorbeeld getoond in een dashboard of een app.

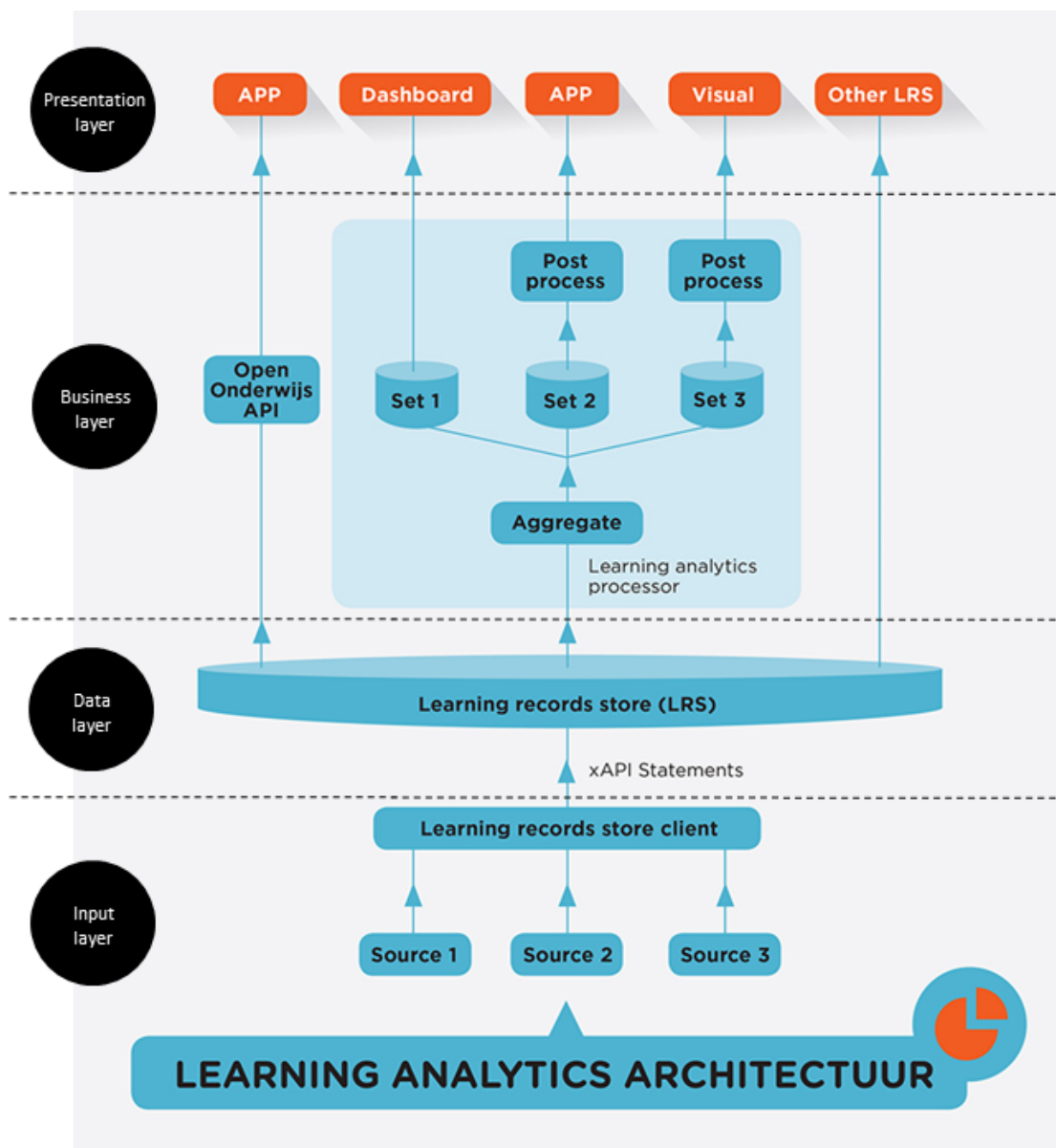
- De business-laag die de functionaliteiten van het experiment levert.

Dit is de Learning Analytics Processor die de data uit de Learning Records Store (LRS) aggregiert, ordent, analyseert en op maat maakt voor de verschillende afnemers uit de presentatie-laag. Naast het gebruik van de LRS door de Learning Analytics Processor kan de LRS ook door andere tools ontsloten worden, zo is er ook een specificatie voor learning analytics in de OOAPI.

- De data-laag, het hart van deze architectuur.

Het belangrijkste component is de Learning Record Store voor de opslag van de studentactiviteiten uit de verschillende online (leer)omgevingen die studenten gebruiken.

- De input-laag, waar verschillende bronnen (omgevingen) gekoppeld worden die de LRS voorzien van activiteiten.



De kenmerken waar de architectuur voor het Learning Analytics experiment aan voldoet, zijn:

- De gebruikte componenten zijn onafhankelijk van leveranciers, zijn vervangbaar en maken gebruik van open standaarden.
- De architectuur staat los van welke bronproducten er worden gebruikt bij de instellingen en past binnen een persoonlijke en flexibele leeromgeving.

Het proces: data verzamelen (Capture)

De input-laag zorgt ervoor dat data verzameld wordt in de LRS. Activiteiten van studenten komen uit verschillende bronnen, die via de door SURFnet ontwikkelde LRS-client op een uniforme wijze in de LRS terecht komen.

Er zijn drie hoofdfunctionaliteiten te onderscheiden voor dit proces:

1. Het tracken van de student. Omdat activiteiten van studenten verspreid zijn over verschillende bronnen, is het belangrijk om de student op een eenduidige manier te identificeren bij iedere bron.
2. Voor de docent wordt het makkelijk gemaakt om activiteiten van de student te verzamelen, dat wordt gerealiseerd door de complexiteit van xAPI uit de bron te halen en met eenvoudige javascript code de activiteiten te monitoren via de LRS-client.
3. De javascript code uit een bron wordt vertaald naar een xAPI-statement die als input dient voor de LRS. Deze xAPI statements leggen eenduidig vast wat de betekenis is van een activiteit van een student, waarvoor eigen xAPI recepten zijn ontworpen.

xAPI is een belangrijke standaard in learning analytics en wordt hier beschreven: <https://experienceapi.com/>. Recepten worden gebruikt om afspraken te maken over het domein van de activiteiten. Een student kan bijvoorbeeld 'ingelogd zijn' of 'aangemeld zijn', waarbij waarschijnlijk hetzelfde wordt bedoeld. Het recept beschrijft de term die gebruikt moet worden.

De voordelen van het inzetten van een LRS-client voor het verzamelen van data in plaats van direct koppelen van bronnen aan de LRS zijn:

- Gebruik van een LRS-client is veiliger, omdat de echte LRS-credentials niet beschikbaar hoeven te zijn bij de bron.
- Eindgebruikers (die data uit bronnen verzamelen) hebben geen kennis nodig van xAPI, maar hen wordt een eenvoudige javascript configuratie gepresenteerd.
- De LRS-client biedt gecontroleerde toegang tot resources en recepten, die hier gedefinieerd worden.

Het proces: data rapporteren (report)

Het data verzamelen resulteert in een grote bak met activiteiten van studenten in de LRS. In de learning analytics processor worden statements van een student geaggregeerd in een dataset die vervolgens als input dient voor een visualisatie van de data. Voor verschillende visualisaties worden verschillende datasets klaargezet.

Datasets worden opgeslagen in een uniform formaat, waar post processing op los gelaten kan worden om data klaar te zetten voor een visualisatie. Eenvoudige visualisaties zullen geen/weinig post processing nodig hebben. Voor andere visualisatie kan het nodig zijn data te interpreteren en op de juiste manier te rangschikken alvorens te visualiseren.

De visualisaties waar specifiek voor het Learning Analytics experiment op gefocust wordt, zijn visualisaties van de vragen die de docent beantwoord wil zien. Een visualisatie dient eenvoudig door de docent te interpreteren te zijn, zodat hij zijn vervolg stappen erop kan baseren. Afhankelijk van de complexiteit van de vraag en daarmee de visualisatie, kan het noodzakelijk zijn dat in de post processing de taal 'R' moet worden gebruikt om de analyse te doen, zie: <https://www.r-project.org/>. Dit moet blijken gedurende de ontwikkeling van het experiment.

De voordelen van de learning analytics processor voor het rapporteren over de data zijn:

- Data wordt aangepast voor specifieke use cases/visualisaties
 - Datasets worden zo klein mogelijk gehouden
 - Grote datasets zijn ook mogelijk voor een 'main' dashboard
 - Aggregatie van data is één proces, met als output meerdere datasets in één uniform formaat
 - Post-processing op een dataset is mogelijk om data klaar te maken voor (meer complexe) visualisaties
-