

# SAKAI PROJECT

## Modularisatie in Sakai

L.P. Verspeek - 10184465  
University of Amsterdam  
Department of Computer Science

21 Januari 2013

### INLEIDING

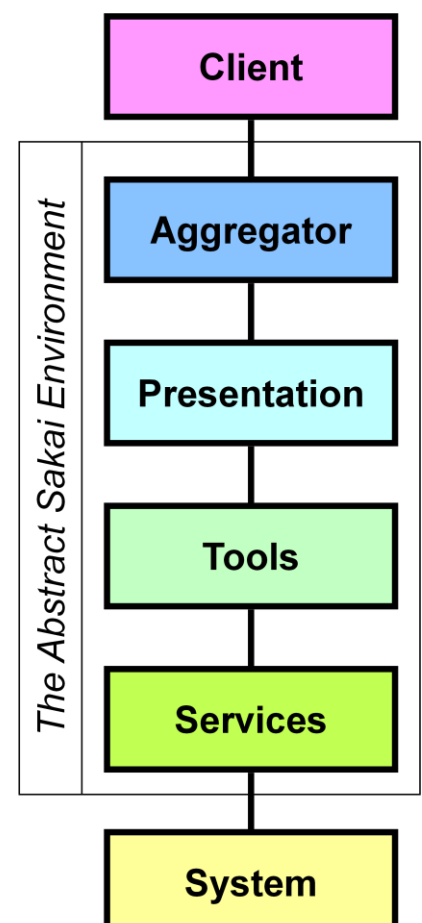
Het Sakai Project is de samenwerking tussen verschillende organisaties voor de ontwikkeling van educatieve open-source software.

In dit verslag wordt eerst gekeken naar de architectuur van Sakai. Vervolgens kijken we naar de voor en nadelen van deze architectuur, met name hoe wordt omgegaan met modularisatie. Ook wordt Sakai vergeleken met een ander open-source project genaamd Eclipse.

### SAKAI'S ARCHITECTUUR

Aangezien het de bedoeling was dat er vele ontwikkelaars tegelijk aan Sakai zouden werken moest de architectuur van Sakai modulair zijn. De architectuur van Sakai is op te splitsen in vier lagen.

1. De aggregator laag zorgt ervoor dat alle tools, knoppen, tabs, etc samen worden gevoegd en produceert de uiteindelijke gebruikersinterface.
2. De presentatie laag zorgt ervoor dat alle tools dezelfde stijl krijgen door elementen aan te bieden voor de interface die hergebruikt kunnen worden.
3. De tools laag toont de uitvoer van de service aan de gebruiker en stuurt de input van de gebruiker door naar de services.
4. De service laag bevat alle logica die ervoor zorgt dat de data verwerkt wordt tot iets bruikbaar voor de gebruiker en wordt gedeeld door alle tools. De services in deze laag hebben een API die beschikbaar moet zijn vanuit andere services en tools.



## **ONTWIKKELAARS**

Aangezien Sakai open-source software is zijn er vele mensen en instanties die aan Sakai werken. Er zijn vier officiële teams die op de Sakai website worden vermeld.

### **Sakai Fellows**

Elk jaar worden er zes Sakai Fellows geselecteerd door een klein comité bestaande uit Sakai gemeenschapsleden. Fellows ontvangen een bescheiden toelage om de kosten van Sakai-gerelateerde activiteiten te dekken. Het doel is om leiderschap en contributies binnen de gemeenschap te bevorderen.

### **The CLE Technical Coordination Committee**

De TCC coördineert de gemeenschap door technisch advies en ondersteuning te geven aan Sakai CLE. Hierin zitten mensen met veel kennis en expertise van Sakai.

### **The CLE Maintenance Team**

Het CLE onderhouds team is een groep van gemeenschapsleden die proberen alle problemen met het Sakai CLE product op te lossen. Deze kritische groep bewaakt de architectuur van het product, repareert bugs, voert patches uit en helpen om de issue tracker te onderhouden.

### **The Sakai OAE Steering Group**

The Sakai OAE Steering Group is verantwoordelijk voor overzicht van het Sakai OAE project en zorgt voor een goede balans tussen lokale behoeften van contribuerende instanties en de vereiste om een succesvolle release van de gemeenschap uit te brengen.

## **VERGELIJKING MET ECLIPSE**

De architectuur van Eclipse is net zoals die van Sakai modulair en open-source. Ook bij Eclipse werken er dus vele ontwikkelaars tegelijk aan hetzelfde product. Toch zijn er wel verschillen tussen beide. Eclipse bestaat uit een basis platform met plugins. Deze manier van bijdragen via plugins is in Sakai niet mogelijk. Dit verschil is ontstaan doordat Eclipse gebaseerd is op de taal waarmee je wilt programmeren en hierdoor wil je als je Eclipse eenmaal hebt draaien makkelijk een plugin erbij kunnen downloaden om bijvoorbeeld een andere programmeertaal te ondersteunen. Sakai zal niet vaak worden aangepast als het eenmaal draait (op de updates na natuurlijk). Een uitbreiding in Eclipse is dus makkelijker toe te voegen dan een uitbreiding in Sakai, maar in Eclipse wil je ook veel vaker aanpassingen dan met Sakai.

## **AFWEGINGEN**

Door modulaire architectuur van Sakai wordt de afhankelijkheid tussen verschillende onderdelen van Sakai zo klein mogelijk gemaakt. Hierdoor is het mogelijk om met veel verschillende ontwikkelaars aan hetzelfde product te werken zonder dat je de hele tijd op elkaar moet wachten.

Zo'n modulaire architectuur heeft ook nadelen. Zo is het erg moeilijk om een modulaire architectuur goed op te zetten, waardoor de software erg complex kan worden. Ook kan het meer werk zijn om simpele tools te ontwikkelen omdat de logica, presentatie en data van zo'n tool apart moeten worden gehouden. In een architectuur met weinig/geen modulariteit werken

de componenten beter samen waardoor de performance beter is dan in een modulaire architectuur, maar de flexibiliteit gaat hierdoor wel verloren. Een modulaire architectuur zoals Sakai heeft kost dus performance, maar is veel flexibiler.

## **BRONNEN**

<http://www.sakaiproject.org>

<https://confluence.sakaiproject.org/display/DOC/Abstract+Architecture>

[http://wiki.eclipse.org/Architecture\\_Council](http://wiki.eclipse.org/Architecture_Council)