

Sunets prototyp

Sunet har utvecklat en prototyp modellerad efter Lunds lösning, med en synkmotor skriven i Python och ett enkelt användargränssnitt för att hantera schemakopplingar. Användargränssnittet är "proof of concept" och består av en extern webbserver som inte är integrerad i Canvas som ett LTI-verktyg. I nuläget består gränssnittet av en lista av kurstillfällen från TimeEdit jämte en lista med Canvas-kurser. Det är alltså ett manuellt gränssnitt och har inte någon koppling till Ladok eller automatiserade kopplingar baserade på tex kurskod.

Idéer på potentiell framtida utveckling

- Generalisera lösningen på TimeEdit- och Canvas-sidan så att kopplingar kan ske mellan vilka två "objekt" som helst. Exempel på objekt på TimeEdit är: Kurs, lokal, studentgrupp, lärare. Dessa kan variera mellan lärosäten. Objekt i Canvas är: Kurs, grupp, användare. Dessa är konstanta.

En sådan här generalisering bör inte kräva mycket utvecklingsarbete. Den skulle förhoppningsvis täcka olika lärosätens behov av manuell schemakoppling väl utan att komplicera saker för mycket.

- Skapa automatiskt schemakopplingar baserat på tex kurskod.

Detta är mer komplicerat och kan vara svårt att implementera generellt. Det skulle möjligtvis kunna göras på någon sorts "best effort" basis där kopplingar skapas manuellt men gränssnittet kan föreslå en koppling baserat på tex kurskod ifall en sådan match finnes. För en helt automatisk lösning krävs nog att vi ställer krav på hur lärosäten strukturerar TimeEdit och Canvas, tex att de använder Sunets existerande [Canvasadapter](#) för att skapa kursrum i Canvas baserat på Ladok.

- Implementera generell import av [iCalendar](#)-flöden till Canvas.

Det här skulle innebära (kanske mycket) mer arbete, men om det gick att göra så skulle det innebära att vi inte behöver köra en egen synkmotor. Det finns [en issue](#) på Canvas från 2015 som efterfrågar detta. I brist på sådan funktionalitet i Canvas verkar de flesta ha löst det med extern synkmotor som vårt test och Lunds universitet har gjort.

Ett problem med den här idén är att även om vi får import av iCalendar implementerat i Canvas, så verkar det inte finnas något bra sätt att få ut iCalendar-länkar från TimeEdits API.

- Integrera gränssnitt för hantering av schemakopplingar i Canvas som LTI-verktyg.

Vad detta innebär för utvecklingsinsats har inte undersökts närmare.

Idéer för drift och underhåll

Integrationen mellan TimeEdit och Canvas är inte kritisk så till vida att något av systemen blir obrukbart om tjänsten går ner och alla data kommer fortfarande att finnas i TimeEdit respektive Canvas. Detta innebär att Sunets IaaS kan användas för att köra tjänsten, till att börja med kommer en instans krävas där hela tjänsten körs. Om behov uppstår i framtiden kan databasen (Postgres) separeras från övriga delar och köras på en egen instans.

För backup så kan Sunet BaaS användas och det som främst bör säkerhetskopieras är databasen som innehåller alla kopplingar mellan TimeEdit och Canvas.

Sunets NOC bör sköta övervakning, och tjänsten är som sagt inte kritisk så en rimlig nivå för övervakningen är att tjänsten monitoreras under kontorstid och att eventuella fel åtgärdas så fort som möjligt men inte utanför arbetstid. Om tjänsten skulle vara otillgänglig en längre tid så finns all data som sagt fortfarande att tillgå i TimeEdit respektive Canvas.