



# Kaliber

Teknisk förstudie

## Bakgrund och syfte

Rapporten beskriver ett nytt system, med arbetsnamn "Kaliber", som ska knyta samman olika bibliotekssystem, e-boksleverantörer och innehållsleverantörer för att bibliotekens besökare ska kunna hitta, låna och läsa e-böcker på ett mycket enklare sätt än vad som idag är möjligt. Rapporten inkluderar en övergripande systembeskrivning samt arkitekturförslag och integrationspunkter.

Ett viktigt syfte är att undersöka om det är tekniskt möjligt att bygga systemet och vara bas för en uppskattning av vad det färdiga systemet kan komma att kosta att bygga och underhålla.

Rapporten är teknisk till sin karaktär, och vänder sig främst till en teknisk målgrupp. Alla läsare bör dock ha behållning av sammanfattning och slutsatser.

## Sammanfattning

I dag har biblioteken hamnat lite på efterkälken vad gäller förmedling av e-böcker. Exempel på eftersatta områden är möjligheten att välja vilka böcker som ska vara tillgängliga och förmedla innehållet, samt svårigheter från ett användarperspektiv att hitta och låna böcker, och att föra över dem till sin läsare.

Kaliberlösningen syftar till att tillgängliggöra e-böcker till bibliotekens låntagare på ett användarvänligt sätt. Förutom att det ska vara enkelt att hitta, låna och läsa böcker så är en viktig målsättning att kunna erbjuda läsoplevelser oavsett böckernas ursprung.

Lösningen skall t ex kunna hantera fria böcker och inte begränsa möjligheterna för exempelvis "self-publishing" i ett senare skede.

De kommersiella villkoren för utlåning av e-böcker via bibliotek har debatterats under en längre tid. Varken Förläggareföreningen eller Biblioteksforeningen har varit nöjda. Framöver kommer det finnas ett antal olika sätt för förlagen, inte minst då flera bokdistributörer kopplas in, att få ersättning för böckerna som lånas. Detta gör att Kaliber måste kunna stödja flera olika licensmodeller och kunna anpassas i takt med att de kommersiella villkoren ändras.

Vi tror att Kaliber i möjligaste mån bör hålla ihop användarupplevelsen och erbjuda en integrerad lösning för att inspirera, finna, låna och läsa e-böcker. I annat fall finns en betydande risk att man tappar låntagare mellan del olika stegen.

En fördel med en integrerad lösning är att biblioteken får möjligheten att lära sig hur låntagarna konsumerar böcker och kan t ex rekommendera fortsatt läsning i samband med att en bok avslutas.

Fördelarna med att hålla ihop användarupplevelsen måste naturligtvis vägas mot kostnaden.

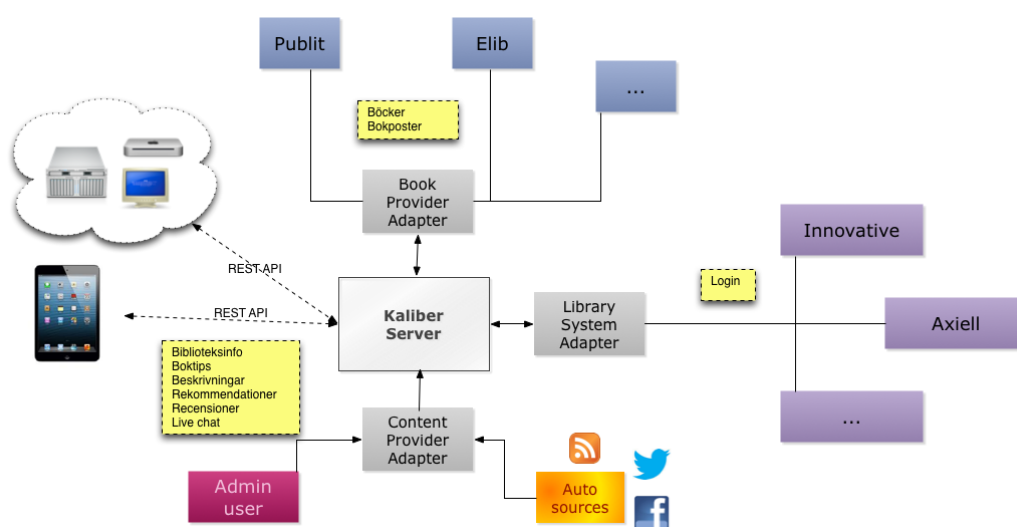
En integrerad och optimerad användarupplevelse kräver kortsiktigt en del mer utveckling än om man bygger ihop- och integrerar systemet genom färdiga komponenter som exempelvis bibliotekssystem, läsare etc.

Rapporten innehåller en genomgång av olika integrationsmöjligheter, ett övergripande arkitekturförslag och avslutas med resurs- och kostnadsuppskattningar samt en rekommendation till lösning och fortsatt arbete.

## Introduktion

Den föreslagna lösningen är en klient/server applikation där klienten, en mobilapplikation (iOS och Android), kopplar sig till servern via ett REST API. Applikationen är till för att låna och läsa e-böckerna.

Redaktionellt Innehåll (recensioner, tips, topplistor etc.) publiceras av de olika anslutna biblioteken för att erbjuda mervärden åt sina lokala användare. Det går också att centralt publicera generellt innehåll som berör alla användare, oavsett vilket som är deras hemmabibliotek. Kaliber knyter ihop olika bibliotekssystem och e-boksleverantörer för att erbjuda en konsoliderad tjänst där användaren får en så bra upplevelse som möjligt av att söka, ladda ner och läsa e-böcker.



*Det övergripande systemdiagrammet visar hur Kaliber binds ihop till de olika delar som beskrivs i denna rapport.*

## Kaliber server

Serverns uppgift är att agera som ett nav och koppla ihop flera olika externa system för att klienten endast skall behöva kommunicera med en kontaktpunkt.

Huvuduppgiften är att hjälpa låntagare att söka, finna och låna böcker. Detta görs genom att erbjuda en katalog med bokposter som låntagarna kan söka i.

För att underlätta för låntagaren att hitta de böcker hon vill ha sparar servern även information om topplistor, rekommendationer, betyg, favoriter och eventuell annan tilläggsinformation. Denna information kan användaren bläddra i, men den kan även användas för att sortera träfflistan i den ordning som passar användaren bäst.

När en låntagare har hittat en bok som hon vill låna kontrollerar Kaliber via det lokala bibliotekssystemet att hon är en behörig låntagare. Denna kontroll behöver bara göras en gång eller så ofta som användaren eller biblioteken vill att den görs.

När en låntagare har loggat in/autentiserat sig mot sitt bibliotek kopplar servern upp sig mot den bokleverantör som erbjuder den aktuella boken. Boken lånas, och en URL (webblänk) skickas till klienten som då laddar ner boken. När lånet genomförts skickas notifikation till det lokala bibliotekssystemet så att det kan uppdatera sin information om så önskas.

Servern kan också hålla information om pågående lån, reservationer och liknande.

Så mycket information som är tillåtet sparas på servern för att underlätta för användaren att använda sig av fler klienter. Exempelvis sparas bokmärken i varje bok som en låntagare läser så att låntagaren kan hitta var man läste senast. På så vis kan man använda valfri klient för fortsatt läsande. Om någon information är känslig och användaren inte vill att den ska sparas på servern finns möjlighet att spara denna information på klienten istället.

Servern kopplar sig mot tre olika typer av externa system:

1. Bokleverantörer
2. Bibliotekssystem
3. Innehållsleverantörer

De olika systemen ansluts med hjälp av olika adapterar för att det ska vara enkelt att byta mellan olika leverantörer. Med adapter menas i sammanhanget en modul som agerar brygga mellan det externa systemet och Kaliber.

### Bokleverantörer

Bokleverantörerna bidrar med två informationsposter; dels bokposter, dels e-böcker.

Bokposterna laddas ner när en leverantör ansluts, och de uppdateras sedan kontinuerligt när förändringar av leverantörens böcker sker. Metod och frekvens för denna uppdatering skiljer sig mellan olika leverantörer.

E-böckerna laddas ner varje gång en låntagare lånar en bok. Information skickas till de lokala systemen när en bok lånas eller lämnas tillbaka.

Det finns oftast regler för hur länge en bok får lånas, eller hur många böcker som får lånas av en användare samtidigt. Vi föreslår att denna information sparas i Kaliberservern.

Samtidigt finns också den här informationen i vissa externa system. En bok som har Adobe DRM-skydd måste först låsas upp innan den kan läsas. Då kontaktas Adobe Content Server och upplåsningen samt "nedräkning" påbörjas. I samma ögonblick kommer även Kaliber att notifieras, så att systemet får kännedom om att bokens giltighetstid har börjat räknas ner.

## Bibliotekssystem

Bibliotekssystemen används endast för att verifiera att en användare är en låntagare hos de bibliotek där hon vill låna böcker.

Bibliotekssystemen notifieras även när lån och återlämning av böcker sker.

Det kan också vara så att det går att låna e-böcker via ett annat gränssnitt, t ex ett biblioteks webbsida. Det finns då två alternativ för att knyta detta till Kaliber. Antingen används Kalibers API:er för att göra lånet på bibliotekswebben (rekommenderas) eller så har bibliotekswebben eventuellt redan integrerat mot t ex API:er hos Elib. Då får Kaliber notifieras om detta via speciella API-anrop som även tar med själva nedladdningslänken till e-boken, så att den går att ladda ner för användaren från Kaliber.

Problemet med det senare alternativet är att Kaliber inte säkert kan veta hur lång tid boken är tillåten att läsas eftersom användaren kanske började läsa (låsa upp) direkt via nedladdningslänken. Det här problemet gäller alla lösningar där användaren får direkt tillgång till nedladdningslänken utanför Kalibers kontext.

Kaliber möjliggör (om det är önskvärt) att en låntagare har ett kaliberkonto för flera lånekort från olika bibliotek för att hantera och organisera sin e-läsning på ett ställe. Kaliber hanterar flera olika bibliotek genom att användaren verifierar varje nytt lånekort med PIN-kod. Därmed vet systemet att användaren får låna. Om användaren inte får låna en bok (t ex beroende på för många lån inom tidsperioden) kommer boken att synas bland sökresultaten utan att det är möjligt att låna den.

Ett eventuellt införande av ett nationellt lånekort kommer att passa väl i lösningen eftersom det redan kommer finnas en samlad bokdatabas tillgänglig. Om ett nationellt lånekort innebär att det går att låna från alla bibliotek innebär det stora fördelar för användaren eftersom alla böcker i systemet kommer vara tillgängliga.

## Innehållsleverantörer

Redaktionellt material kommer till stor del att ligga i externa system (d v s hos innehållsleverantörer). Motivet till detta är att det ska vara enkelt för biblioteken att lägga till material via system som de redan idag har tillgång till.

Det kommer att finnas standard adaptrar för de vanligaste typerna av informationsflöden såsom RSS och Twitter. RSS är ett format som bland annat WordPress exponerar. Det betyder att innehåll enkelt kan skapas i WordPress (eller annan liknande publiceringsplattform) och nås ifrån Kalibers server för att till sist presenteras för låntagarna.

Nya innehållsleverantörer kan skapas och läggas till utan att övriga delar i systemet påverkas.

## REST API

Servern publicerar ett REST API som klienten använder sig av för att kommunicera med servern. Detta API är öppet så att andra klienter eller system kan ansluta sig till Kaliber. På så sätt blir det möjligt för bibliotek att inkludera information från Kaliber på sin egen webbsida.

## Administrationsgränssnitt

Servern administreras via ett webbgränssnitt. Det finns två typer av administratörer:

- Övergripande systemadministratör - konfigurerar initialt och ansluter bibliotek till tjänsten
- Biblioteksadministratör - för varje bibliotek som ställer in exempelvis:
  - Vilka böcker som skall finnas tillgängliga, inklusive hantering av eventuella dubletter
  - Regler för utlåning
  - Källor för redaktionellt innehåll

## Sekvensdiagram

Sekvensdiagram för några av de viktigaste flödena i Kaliber finns i Appendix II – sekvensdiagram.

## Summering

Kalibers serverkomponent är främst tänkt att användas för att knyta samman bibliotekssystem, e-boksleverantörer och redaktionellt innehåll så att det går att nås via en klientapplikation. Alla API:er som servern exponerar åt klienten är öppna och kan potentiellt användas av andra typer av system.

Exempelvis skulle ett biblioteks webbplats kunna visa information från Kaliber. Det går också att använda en extern "hub" så länge det går att integrera mot. I det fallet agerar Kaliber enbart brygga för att exponera ut API:er till klienten.

## Klienten

På mobila enheter går det att installera en klientapplikation som är slutanvändarens primära kontaktväg gentemot Kaliber. Klientapplikationen är i första hand tänkt att fungera för Apple iPad (iOS) och Android-enheter. I applikationen går det att komma åt Kaliber och söka/finna e-böcker samt låna och läsa dessa.

Målsättningen är att klienten ska vara så "tunn" så möjligt, vilket innebär att så mycket information som möjligt sparas på servern. Detta underlättar för användaren att ha flera Kaliber-applikationer på olika enheter, synkroniserade med samma information. Det ger en bättre arkitektur. Vi möjliggör att spara information antingen på servern eller i klienten baserat på gällande lagstiftning, då det kan vara känsligt att spara personlig information på en server.

Tanken är att klienten är en hybrid-applikation (dvs. en applikation där en del av innehållet är webbinnehåll). Valet av en hybridlösning baseras på en kortare utvecklingstid samt möjligheten att styra innehåll och utseende utan att behöva ladda upp en ny klient till AppStore eller Google Play.

## DRM-läsare för mobil enhet

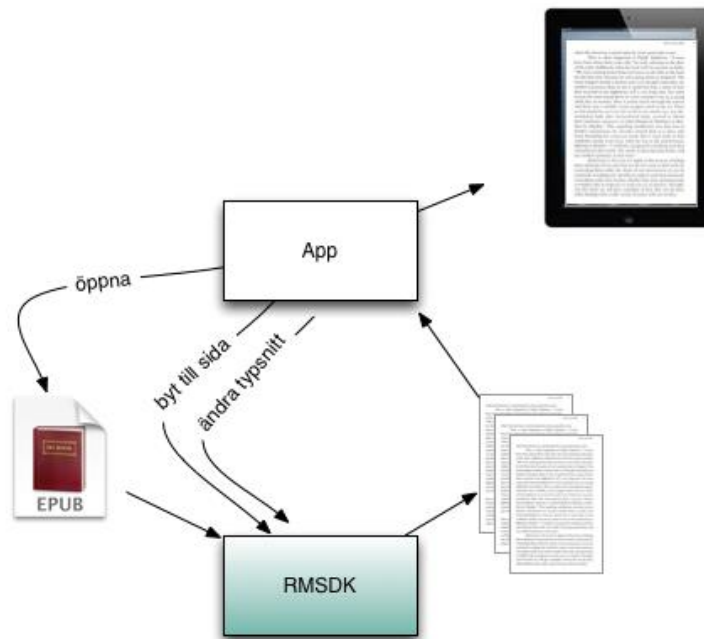
DRM står för Digital Rights Management och syftar på system som, för att skydda upphovsrättsinnehavarens intressen, begränsar hur en produkt får användas. I fallet e-böcker rör det sig oftast om att en bok ska kunna lånas och läsas under en begränsad tidsperiod, eller att en bok köps och bara kan öppnas av ägaren. Detta för att förhindra piratkopiering.

## Böcker skyddade med Adobe DRM

För att kunna läsa e-böcker som är skyddade med DRM från Adobe krävs speciell mjukvara som hanterar detta. Bokleverantörer som ELIB använder systemet för att kunna begränsa så att användare bara får lov att läsa en lånad bok under en viss tidsperiod.

Den enda mjukvara som kan öppna Adobe-DRM-skyddade e-böcker är Adobe Reader Mobile SDK (RMSDK). Denna produkt finns för flera plattformar, bland annat iOS och Android. RMSDK är ett bibliotek som ger ett utvecklargränssnitt för att hantera e-böcker i ePub- och PDF-format. ePub3-formatet stöds ännu inte av RMSDK men det pågår intensivt arbete för att det ska implementeras.

För att kunna öppna en Adobe-DRM-skyddad bok behöver användaren normalt ett Adobe-konto med login och lösenord. Undantaget är om man använder Adobe Vendor ID, vilket är en tjänst från Adobe för att göra upplevelsen smidigare för slutanvändaren. Det innebär i praktiken att användaren inte behöver skapa ett eget Adobe ID. Istället hanteras autentisering via Kaliber-servern som har ett enda Vendor ID som används för alla kunder. Det är mycket önskvärt eftersom det eliminerar ett stort hinder för användarvänligheten. Kostnaden för denna tjänst är ca \$35 000 per år.



En app som använder RMSDK behöver bara skicka in en ePub/PDF till RMSDK för att få tillbaka en färdigrenderad sida i png-format. Den här bilden får appen sen visa för användaren. När användaren vill byta sida kommer appen att begära en ny sida från SDK och en ny sida renderas.

Här följer en lista som exemplifierar ansvaret för app respektive RMSDK:

#### Appens ansvar

- Hantera och spara länk (lokal eller remote) till e-bok
- Felhantering och notifieringar
- Synkning av senast läst sida
- Användargränssnitt för funktioner som byt sida, ändra fontstorlek mm
- Konfigurera RMSDK för aktuell enhet, t.ex. skärmstorlek och rotation

#### RMSDKs ansvar

- Ladda ner och öppna e-böcker
- All hantering av DRM
- Hantera allt innehåll i e-böcker
- Funktion för att rendera bild av en viss sida
- Acceptera konfiguration av hur en sida ska visas

Trots att appen bara känner till bilder kan viss interaktivitet med e-boken kan möjliggöras, t ex genom följande exempelscenario:

#### Scenario: en användare vill markera text och skicka till Facebook som ett citat

- Användaren drar på skärmen för att markera ett citat
- Appen skickar ett meddelande till RMSDK: "Användare markerade text med position (x,y), (x2,y2)"
- RMSDK svarar appen med en textsträng som motsvarar det användaren markerade
- Appen skickar textsträngen till Facebook



## Böcker som är oskyddade

Många böcker, bland annat från Gutenberg, har inget DRM-skydd utan är helt fria att ladda ner och läsa som användaren önskar. Därför behövs inte heller någon speciell hantering i bokläsaren. RMSDK från Adobe klarar dock av att öppna och läsa oskyddade böcker på samma sätt som skyddade böcker. Därför finns ingen anledning att ha en speciell lösning bara för gratisböckerna.

## ”Streamade” e-böcker

En variant av läsning av e-böcker är att böckerna inte alls laddas ner till klienten, utan varje sida hämtas direkt från Internet då den behövs/ska läsas. Ett exempel på en sådan tjänst är Elibs ElibU.

Det finns flera fördelar med detta: e-böckerna behöver inte ta lagringsutrymme lokalt, intressanta affärsmodeller (som t.ex. att biblioteken ersätter förlagen per läst sida, istället för per bok) möjliggörs; och inte minst behövs ingen läskomponent i appen.

Nackdelar med detta förfarande kan eventuellt vara problem med att läsa e-böcker utan nätuppkoppling, begränsade möjligheter att påverka läsarens utseende och funktionalitet, samt osäker kompatibilitet med olika e-bokformat.

## Alternativ egen DRM-lösning

Det finns flera nackdelar med DRM från Adobe:

- Det krävs implementation för varje plattform som ska stödjas, vilket kostar
- Det krävs en årlig återkommande licensförnyelse med Adobe
- Det går långsamt att innovera eftersom Adobe måste leda utvecklingen med ny funktionalitet
- Vissa plattformar, t.ex. Linux, stöds inte

Ett alternativ, som kanske inte är aktuellt i dagsläget, kan vara att avstå från Adobe DRM och istället bygga en egen lösning. Det handlar då om att tillgodose både upphovsrättsinnehavarnas önskemål om säker hantering som försvårar piratkopiering, och att skapa en så smidig och bra användarupplevelse som möjligt.

Detta skulle kräva att förlagen går med på att leverera oskyddade e-böcker till vårt system, för att där förses med ett skydd. Det skulle också göra det mindre intressant att integrera med leverantörssystem annat än för att importera e-böckerna.

## Summering

Den bästa lösningen med tanke på användarvänlighet och möjligheter är att licensiera Adobes RMSDK och använda Vendor ID. Det här ger full flexibilitet att utveckla en bokläsare precis efter bibliotekens önskemål och formspråk, gott och beprövat stöd för olika varianter och format av e-böcker och användaren behöver inte bekymra sig om att skapa ett Adobe ID. Det är också det dyraste alternativet där det finns en löpande kostnad per år samt utvecklingskostnad att bygga en egen skräddarsydd bokläsare och underhåll av denna.

## Anslutning av nya system till Kaliber

Detta avsnitt beskriver hur nya system integreras med Kaliber och vad som krävs vid respektive integration.

### Att ansluta en ny bokleverantör

För att ansluta en ny bokleverantör behöver det skapas en adapter. Detta kräver utvecklingsarbete i Kaliber server, men kräver inte några ändringar alls i klienten. Adaptern innebär integrationsarbete mot det specifika API som bokleverantören tillhandahåller. Ur resten av Kaliber kan sen innehållet nås på ett standardiserat sätt.

Adaptern abstraherar alltså bort den potentiellt komplexa integrationen. När väl en ny bokleverantör har en adapter kan biblioteken välja att ansluta till denna med sin anslutningsinformation (se nedan).

### Att ansluta ett nytt bibliotekssystem

På samma sätt som för bokleverantörerna behövs det endast en adapter för det nya bibliotekssystemet. Adaptern innehåller den logik som behövs för att kommunicera med det specifika API:et. Kaliber i övrigt påverkas inte.

Kaliber är främst tänkt att användas för att hantera e-böcker, men det är inte omöjligt att även kunna visa upp information om fysiska böcker. Ur ett tekniskt perspektiv kräver detta mer kommunikation mellan Kaliber och de lokala bibliotekssystemen. Detta är inte önskvärt men görbart. Förslagsvis hämtar Kaliber låneinformation på regelbundna intervaller ifrån de lokala systemen.

### Att ansluta en ny informationsleverantör till tjänsten

Om man vill ansluta en leverantör som stöds av en standardadapter behöver man bara tillhandahålla anslutningsinformation (API endpoint). Standardadapterar är RSS, Twitter, etc.

Om man vill ansluta en ny typ av information så krävs en ny adapter. Detta medför utvecklingsarbete på serversidan men innebär troligtvis ingen förändring i klienten (såvida inte det ska visas upp på ett nytt sätt).

### Att ansluta ett bibliotek till tjänsten

När ett bibliotek ansluts **måste** det tillhandahålla:

- Uppgifter för hur biblioteket ansluter till bokleverantörer om biblioteket har ett eget konto.
- Information om anslutning till bibliotekssystemet (API endpoint) för att en användare ska kunna logga in.

Exakt vilken typ av information som behövs beror på vilka bibliotekssystem och bokleverantörer som används. Se exempel om Malmö Stadsbibliotek nedan.

Det är **önskvärt att** systemet kan tillhandahålla:

- Anslutningsinformation till informationsleverantörer - om inte visas bara det allmänna informationsflödet som tillhandahålls av Kaliber.

### **Flöde:**

Se även sekvensdiagram "Ansluta nytt Bibliotek"

När anslutnings-information finns tillgänglig kommer Kaliber att kontakta de olika leverantörerna för att hämta informationen.

Från bokleverantören hämtas de bokposter som biblioteket prenumererar på. Posterna indexerar och läggs in i Kalibers katalog.

Från informationsleverantörerna hämtas information som sedan visas när låntagare för det biblioteket ansluter sig.

## **Exempel, anslutning av Malmö stadsbibliotek**

Följande behövs för att ansluta biblioteket:

### **Bibliotekssystem:**

Innovative används och Låntagare verifieras via ett Patron API. Anslutningsnycklar för API:et behövs

### **Informationsleverantör:**

Separat Kaliber Wordpress RSS. Adress till RSS-feeden behövs samt eventuellt användarnamn/lösenord.

### **Bokleverantör:**

Malmö har Elib och Overdrive som bokleverantörer:

#### **Elib:**

- Kontrakt
- retailerid
- retailerkeycode

#### **Overdrive:**

- Kontrakt
- API key
- library id

## **Summering**

Att koppla in nya system innebär utveckling av en adapter, men påverkar inte Kaliber i övrigt. Det här ger en starkt modulariserad arkitektur som gör att integrationer kan utvecklas isolerat och även läggas till och ta bort helt dynamiskt. Kaliber-servern kommer troligtvis (beroende på driftmiljö) inte ens behöva startas om då ett nytt system kopplas in.

Det enda som ett bibliotekssystem måste exponera är ett sätt att verifiera att en användare är en låntagare. Alla system har denna möjlighet. De flesta har ett API och resten har en webbsida som det går att utnyttja.

Om systemet vill hålla reda på sina låntagares e-boklån kommer Kaliber att exponera alla lån via sitt API. Detta kan antingen ske via anrop till det lokala systemet eller via ett RSS-flöde som biblioteket kan koppla sig mot. Det går även att skicka denna information via Email, etc. All information skickas krypterat över HTTPS.

Genom anrop till Kalibers API kan biblioteken kunna visa låneinformation för användare på sin egen webbsida.

## Externa System

Kapitlet tar upp de system som upplevts mest relevanta av bibliotekssystem och boksystem.

### Bibliotekssystem

De bibliotekssystem som vi har valt att undersöka är vanliga system på den svenska marknaden. Det vi främst tittat på är möjlighet att autentisera användare via API, samt teknisk komplexitet i att integrera mot systemet.

#### **Axiell BOOK-IT (7.1)**

BOOK-IT använder sig av XML-baserade web services. De är mycket omfattande och har stöd för det mesta. Det är noga dokumenterat, men dokumentationen är också mycket omfattande vilket kan medföra att det tar tid för en integratör att sätta sig in i hur gränssnittet fungerar.

#### **Axiell LMS Web Services**

Består av en samling av SOAP (Simple Object Protocol)-tjänster. Det här är en standard som används mindre och mindre eftersom det är förhållandevis komplext att integrera mot.

#### **Innovative's Patron API**

Har ett mycket naivt API som bara kan hämta "record" för en låntagare (en sorts konto-status) och kontrollera dess PIN-kod. Skickar tillbaka data i ostrukturerad "HTML".

Det här gör att standardverktyg inte kan användas för att kommunicera med API:et utan det kommer ta längre tid än det borde att skriva integrationen för Kaliber. Detta är dock en engångsinvestering.

#### **Innovative Sierra Web Services**

Innovative Sierra använder en RESTful API-lösning med JSON som databärare över HTTP. Detta är mycket bra och precis så vi vill ha det. Det möjliggör att standardverktyg kan användas för att kommunicera.

#### **CS Library**

Agerar som ett lager ovanpå bibliotekssystem och använder Open Library Access för att knyta samman dem. Kan potentiellt användas som motor i Kaliber, men nackdel att det gör systemet plattformsbaserat till Windows eftersom det använder WCF som byggsten.

#### **Bibliotekscentrum Micromark**

Använder Micromark Open Library Access (MMOLA) vilket i sin tur använder Open Library Access (OLA).

## E-bokleverantörer

Olika leverantörer tillhandahåller e-böcker för förlagens räkning. De har dels en databas med information om böcker och dels lagring av de faktiska böckerna. I de fall då böckerna behöver skyddas av upphovsrättsliga skäl skapas egentligen unika "kopior" av böckerna åt låntagarna, vilket gör att det exempelvis kan visas en digital vattenstämpel inuti boken i stil med "Den här boken är lånad av Ann Jansson". Kaliber håller konsoliderad information om vilka böcker som finns tillgängliga och uppdaterar regelbundet den här informationen från leverantörerna.

### Elib

(version 2.1 av Library API)

API tillgängligt här: <http://www.elib.se/webservices/>

API är baserat på SOAP eller HTTP POST. Vi föredrar det senare eftersom det finns bättre verktygsstöd och är lättare att skriva tester för. Dokumentation är ganska grundläggande. Det finns stöd för att lägga lån, men också autentisering av användaren via lånekortsnummer och kod. Det går också att i förväg veta om en bok går att låna. Om ett bibliotek är på väg att nå sin satta totala begränsning så kommer Elib höra av sig via mail, men i kommande version av API:et kommer det kunna hanteras där istället. Kaliber har inget problem med endera lösningen.

Elib arbetar med en ny version av sitt API, men troligtvis kommer det inte att innebära några problem. Möjligtvis kommer det att innebära att Elib-adaptorn måste uppdateras. Detta kommer bara att påverka servern och inte klienten.

Användare kan utnyttja sitt lånekortsnummer och PIN-kod för att låna böcker via Elib, men egentligen behövs inte det sett ur Kalibers perspektiv. Det beror snarare på vilket bokurval som används och vem som ska betala för lånet.

Slutsats: Inga uppenbara problem att integrera.

### Publit

Publit har ett bra, modernt RESTful API som returnerar XML. Autentisering sker via Basic HTTP Authentication över SSL där apiusername och apipassword skickas med i varje anrop. Publit har stöd för lån via CreateShopOrder-anropet.

Slutsats: Inga uppenbara problem att integrera.

### Overdrive

Använder ett RESTful API som returnerar JSON. API:et är baserat på "hypermedia"-design. Vilket betyder att innehåll har länkar till djupare innehåll. Det här innebär en väldigt flexibel och robust lösning där API:et kan förändras utan att klienterna behöver göra det. Autentisering sker med OAuth 2.0.

Det går bland annat att se lånestatus för en bok, t.ex. hur många "exemplar" som är tillgängliga för utlåning.

Dokumentation finns här:

<https://developer.overdrive.com/docs/getting-started>

Slutsats: Inga uppenbara problem att integrera.

### Gutenberg

Går att få ut information om alla böcker i XML-format. De har endast gratisböcker, så varje bok har en nedladdningslänk i valfritt format.

Slutsats: Inga uppenbara problem att integrera.

### **Litteraturbanken**

Fria böcker. Har inget riktigt API men hela katalogen kan nås via en URL:

<http://litteraturbanken.se/urllista/>

Den här sidan kan sen bearbetas med ”screen-scraping”, vilket innebär att websidan i sig hämtas (istället för ett API-anrop som returnerar data) och enligt ett antal fördefinierade regler söks igenom och filtrerar ut relevant information. ”Screen-scraping” är känsligare för förändringar och kan sluta fungera om t.ex. layout förändras väsentligt.

Ett annat alternativ är att istället importera hela databasen och e-böckerna i sig. Detta får då göras kontinuerligt, t.ex. varje natt, för att vara någorlunda i synk med källan.

Slutsats: Inga uppenbara problem att integrera, men lösningen kan bli instabil.

### **Libris**

Fria böcker. Tillhandahåller API för att söka efter titlar. Det går att använda detta API för att t.ex. få ut information i JSON-format, vilket är utmärkt.

Exempel ”query” för att få ut gratis böcker:

<http://libris.kb.se/xsearch?d=libris&mat:free&format=json>

Slutsats: Inga uppenbara problem att integrera.

### **Runeberg**

Fria böcker. Saknar API. Möjlighet finns att komma åt data via ”screen-scraping”.

Slutsats: Inga uppenbara problem att integrera, men lösningen kan bli instabil.

### **Feedbooks**

Gratis böcker på engelska. Hela katalogen finns tillgänglig som ett feed på:

<http://www.feedbooks.com/catalog.atom>

Slutsats: Inga uppenbara problem att integrera.

### **Self publishing**

Det finns också möjlighet att låta Kaliber agera bokleverantör genom att tillhandahålla en databas dit bibliotek kan ladda upp länkar till public domain-böcker. Det är alltså fråga om öppna böcker som inte behöver skyddas med DRM. Tanken är primärt att biblioteken själva lagrar själva bok-filerna, men Kaliber får information om att böckerna finns så de dyker upp i sökresultat och går att bläddra fram. Böckerna hanteras som alla andra datakällor i Kaliber.

Slutsats: Är ingen integration i sig utan en del av implementationen av Kaliber som system.

### **KBs “e-boksmaskin”**

Skulle LIBRIS utökas med stöd för e-böcker, på det sätt det beskrivs i KBs förstudie “Sammanhållet nationellt system för distribution av e-böcker till bibliotek”, skulle all integration mot e-boksleverantörer ovan vara redundant. Detta skulle göra Kalibers komplexitet mindre samt minska utvecklingstiden markant. Då det inte finns någon tidsplan för när en sådan lösning skulle kunna realiseras förutsätts tillsvidare att Kaliber får innehålla motsvarande funktionalitet.

## Sammanfattning

Följande lista sammanfattar de olika leverantörernas möjlighet till att exportera data om böcker (katalog), möjlighet att genomföra lån av e-böcker, se status för en bok så systemet vet om den går att låna och om leverantören bedöms använda en god säkerhetslösning.

System	Bokexport	Lån	Bokstatus	Säkerhet	Kommentar
Elib	Ja	Ja	Ja	Ja	
Publit	Ja	Ja	Nej	Ja	
Overdrive	Ja	Ja	Ja	Ja	
Gutenberg	Ja	Gratis	Gratis	Nej	
Litteraturbanken	Nej	Gratis	Gratis	Nej	
Runeberg	Nej	Gratis	Gratis	Nej	Ej e-pub, html och text
Libris	Ja	Gratis	Gratis	Nej	
Feedbooks	Nej	Gratis	Gratis	Nej	
Self publishing	Ja	Nej	Ja	Ja	

## Summering

De externa system som listas i det här kapitlet kommer gå att integrera mot, men det kommer ta olika lång tid beroende på teknik, tillgänglig dokumentation och samarbetsvilja hos tredje part. Rekommendationen är att börja med ett fåtal vanliga system, för att sedan fortsätta att integrera fler då det visat sig finnas intresse och ekonomiska incitament. Ur ett tekniskt perspektiv föredras REST-baserade API:er då en stor mängd standardverktyg kan utnyttjas för kommunikationen.

Det är en risk att "samma" bok kan finnas hos flera leverantörer. Detta löses genom automatisk matchning av författare och titel, eller ISBN. Det ska även gå att markera eventuella dubletter i Kalibers administrationsverktyg samt konfigurera vilken prioritetsordning de olika leverantörerna har.

## Tids- och kostnadsuppskattning

Det är givetvis svårt att ge exakt uppskattning av hur mycket arbete som krävs för att utveckla lösningen. Svårigheten består till stor del på att avgränsningarna i dagsläget inte är klara (vilka bokleverantörer och bibliotekssystem ska stödjas, hur mycket funktionalitet ska läsaren, vilken data ska systemet kunna leverera etc.). Detta avsnitt är tänkt att ge en indikation på vad en komplett lösning skulle kunna kosta att utveckla. Uppskattningen baseras på erfarenheter från liknande lösningar.

### Utveckling

Uppskattningen gäller ett system bestående av en komplett serverlösning som driftas i molnet (t.ex. Amazon Web Services), en webb del (både för låntagare och för administration av systemet) samt en iOS- (iPhone och iPad) och en Androidklient (telefon och tablet). Klienterna innehåller en förmedlingsdel och en läsardel och stödjer Adobes DRM-lösning

Vidare bygger uppskattningen på att 2-3 bokleverantörer och 2-3 bibliotekssystem stöds. Ytterligare bokleverantörer tar i genomsnitt en manmånad att integrera.

När man licensierar Adobe DRM ingår mobila referensapplikationer, vilka är tänkta att ligga till grund för läsardelarna i klienterna. Detta gör att utvecklingstiden kan hållas nere och fokus kan läggas på Kaliberspecifika delar.

Uppskattningen är att hela lösningen tar mellan 5000h och 8000h att utveckla och att de mobila klienterna står för mindre än hälften av den totala tiden.

Till utvecklingstiden tillkommer licenskostnaden för Adobes DRM-lösning. Licenskostnaden ligger på \$60 000 första året och därefter \$20 000 årligen.

### Drift och underhåll

Vi rekommenderar att man driftsätter tre olika miljöer i molnet: dev, stage, prod. Detta för att vara säker på att förändringar är vältestade och stabila innan de når riktiga användare.

Några saker som bör krävas av serversystemet för att hålla god tillförlitlighet och prestanda:

- Lastbalansering
- Autoskalning på web frontends
- En storage service som t.ex. DynamoDB eller MongoHQ
- Köer för att få robusthet och lösare koppling mellan frontend och backend
- Workers ska jobba mot köerna, via en delad cache, tex Memcached eller Redis
- Övervakning och larm, t.ex. med New Relic
- Backup och loggning, t.ex. via S3
- Stöd för data mining och rapporter
- CDN, med notifieringar med email och SMS

En grov kostnadsuppskattning för den rena driften är 50 000 SEK per år, plus kostnad för en driftansvarig på deltid.

### Licenskostnader

Om man väljer att använda Adobe "Vendor ID" för att underlätta för låntagarna tillkommer en årlig licenskostnad på \$35 000.

Att licensiera Adobes DRM-lösning för de mobila klienterna kostar \$20 000/år (som ovan under "Utveckling").

Utöver Adobes DRM-lösning så kan resten av lösningen tas fram med licensfri mjukvara.



## Underhåll

Systemet kommer troligen att kräva en hel del underhåll och vidareutveckling under det första året då nya bibliotek läggs till och processer och rutiner sätter sig. En uppskattning är att man bör klara sig på mindre än 1000h, för att klara underhåll och visst fortsatt arbete, det första året och detta behov redan är halverat nästkommande år.

## Summering

Baserat på den arkitektur som beskrivs i rapporten och de antaganden som gjorts är kostnadsestimaten:

**Kostnaden för utveckling av systemet ligger mellan 4,5 och 7,5 miljoner kronor.**

**Kostnaden för drift och underhåll ligger runt 1,5 miljoner kronor första året för att sedan minska markant efterföljande år.**

Estimaten baserar sig på en kostnad av runt SEK900 per arbetad timme.

## Analys och slutsats

Det står klart att Kaliber avsevärt skulle förbättra låntagarnas upplevelse av att inspireras, söka, finna, låna och läsa e-böcker. Bibliotekens möjlighet att förmedla böcker och kunskap till låntagaren skulle också stärkas markant.

Ur låntagarens perspektiv är en sammanhållen tjänst klart att föredra jämfört med en tjänst som förlitar sig på en tredjepartsläsare. Detta måste givetvis vägas mot kostnaden för utveckling och underhåll av lösningen.

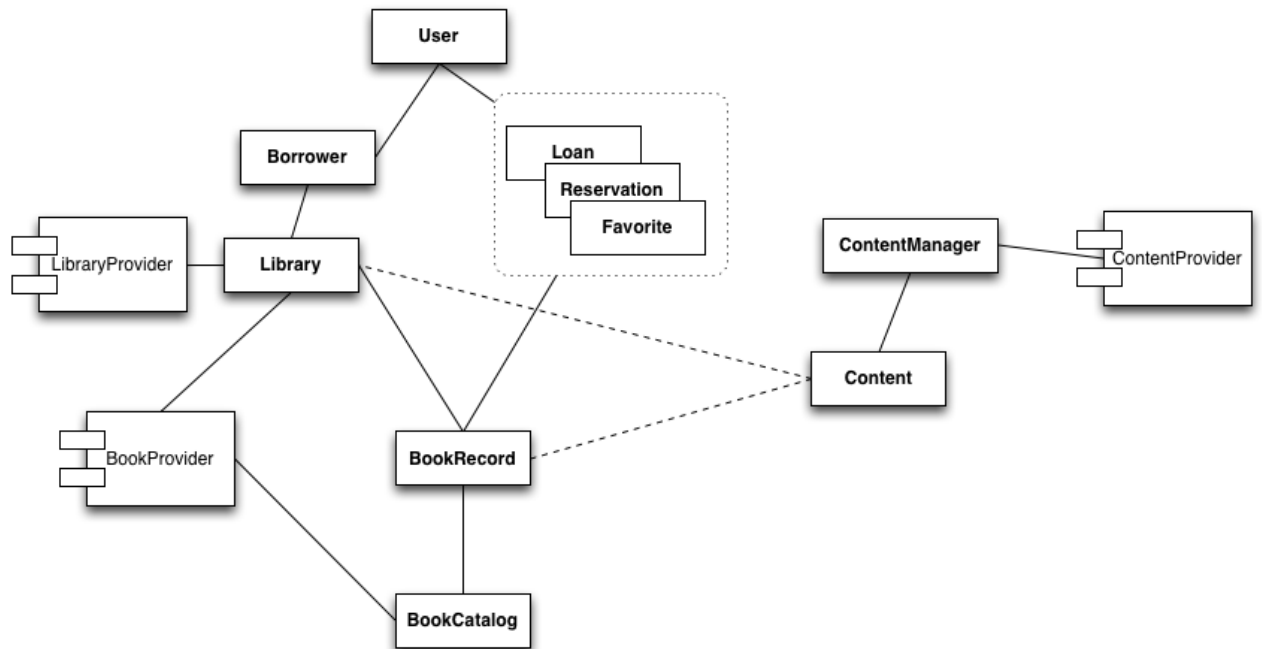
Enligt vår genomgång av olika bibliotekssystem och e-boksleverantörer bedömer vi det fullt möjligt att integrera dessa och koppla dem till Kaliber. Integrationsarbetet blir olika stort för olika leverantörer, och säkerligen finns det idag okända fallgropar; men vi tror ändå det är realistiskt och genomförbart. Det går också att välja i vilken skala tjänsten lanseras. Kanske en första version av Kaliber endast har stöd för ett eller två bibliotekssystem och e-boksleverantörer, för att sedan växa då fler och fler bibliotek ansluter sig.

Vi rekommenderar att man använder så mycket beprövad Open Source (öppen källkod) mjukvara som möjligt för att bygga systemet. Det går att använda sig av välbeprövad och robust kod samtidigt som kostnaden sänks markant. Genom att drift- och leveranssätta systemet i molnet sänks kostnaderna ytterligare (både initialt och då systemet behöver skala), samtidigt som säkerhet och tillgänglighet höjs. Det är också fullt tänkbart att hela Kaliber-systemet, i sig självt, släpps som Open Source. Detta skulle uppmuntra fler parter att vara delaktiga i att göra förbättringar och nya funktioner.

Det finns idag ett tillräckligt underlag för att kunna starta utvecklingen av Kaliber i det närmaste omgående. Från ett utvecklingsperspektiv skulle nästa steg vara att avgränsa systemet och ta fram en prototyp för användartester och för att undanröja tekniska risker. När detta är klart kommer det gå att ta fram en mer detaljerad tidsuppskattning.

En uppskattning av hur lång tid det tar att utveckla lösningen är 6-8 kalendermånader från start till driftsatt system.

## Appendix I - Arkitekturbeskrivning



De viktigaste klasserna i modellen är *User*, *Library*, *BookCatalog*, *ContentManger*.

### User

En *User* är en användare av systemet. En *User* kopplar sig till *Library* via en *Borrower*. En *Borrower* håller reda på det som är speciellt för användaren för just detta bibliotek. En *User* kan vara *Borrower* på flera olika bibliotek.

En *User* hanterar *Loan*, *Reservation*, och *Favoriter*.

### Library

Ett *Library* håller information om ett biblioteks anslutningar. Det innebär hur Kaliber ska koppla sig mot *LibraryProviders*, *ContentProviders* och *BookProviders*.

### BookCatalog

*BookCatalog* innehåller alla poster, *BookRecord*, som finns i systemet. Varje *BookRecord* vet vilken *BookProvider* den kommer ifrån och vilket *Library* den tillhör.

### ContentManager

*ContentManager* hämtar innehåll från olika *ContentProviders*, den sammanställer det och levererar det till klienten baserat på vilket *Library* klienten tillhör.

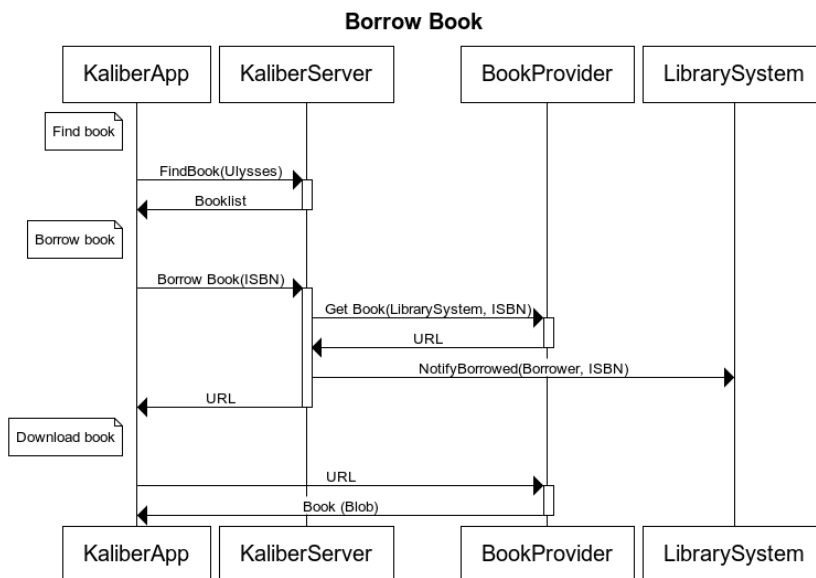
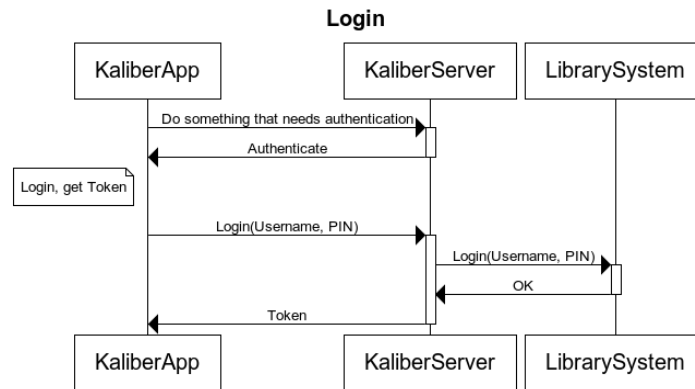
### ContentProvider

En *ContentProvider* hämtar data från en ett externt CMS. Om data hämtas ett definierat format består uppgiften endast i att verifiera att formatet följs och sedan skicka det vidare till *ContentManagers*.

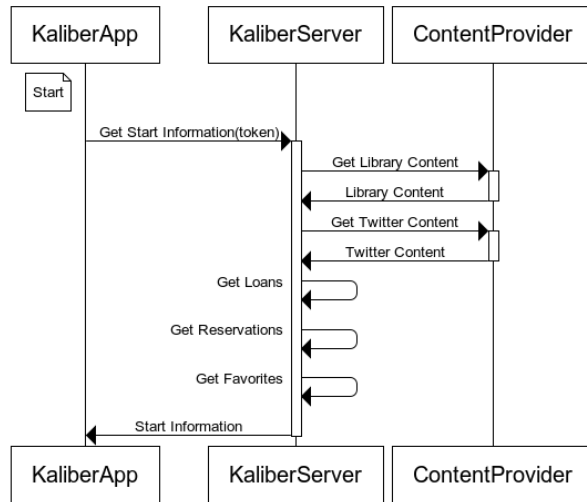
Om innehållet inte följer något speciellt format måste *ContentProvidern* konvertera formatet till ett gemensamt format.

## Appendix II – Sekvensdiagram

Sekvensdiagram för några av de viktiga flödena i Kaliber.



### Start Application



### Ansluta nytt bibliotek

