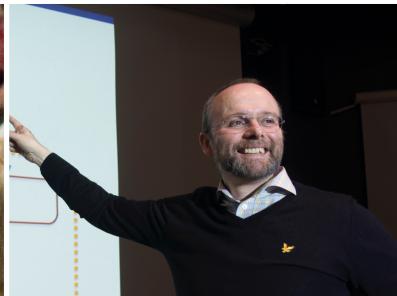


# CADEC 2016

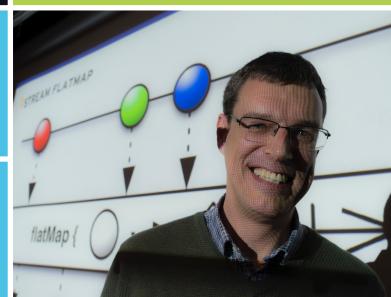
CALLISTA DEVELOPER'S CONFERENCE DEN 27 JANUARI 2016



KUBERNETES  
CLOUD DEPLOYMENT

JAVA 9  
SPRING BOOT

GRAPH QL  
SPRING XD



MICROSERVICES  
DOCKER



WEB COMPONENTS

+ MYCKET MER...

VI BJUDER PÅ DE SENASTE TRENDERNA INOM IT-ARKITEKTUR SAMT AGIL & ÖPPEN SYSTEMUTVECKLING

Även 2016 kommer att starta med en rykande aktuell Cadec som kommer att hållas på IHM i Gårda, Göteborg, precis som tidigare. Konferensen är kostnadsfri.

Eftermiddagen fylls som vanligt med ett gediget konferensprogram som ni hittar nedan. Missa heller inte att varva ner efter konferensen med mat, dryck och mingel på After Cadec.

Varmt välkommen till en dag med kompetensutveckling!

Detaljerad information och anmälan på vår hemsida

[callistaenterprise.se/cadec2016/](http://callistaenterprise.se/cadec2016/)

KONFERENS  
13.00 - 18.00



## WEB COMPONENTS

SEDINA ORUC

För 3 år sedan var WebComponents en stor nyhet, som lovade mycket. Målet var att etablera en samlad bild på specifikationer för: "CustomElements", "HTML Imports", "Templates" och "Shadow DOM". Ramverken Ember och Angular planerade för att integrera dem, till och med basera sig på dem.

Vad är det som har hänt under tiden? Är det fortfarande ett aktuellt ämne? Hur långt har man kommit med dessa specifikationer? Dessa frågor ska jag försöka att besvara under den här presentationen samtidigt som jag ska försöka dra några paralleller med dagens favoritramverk som Reactjs och Angular.

## GRAPH QL

STEPHEN WHITE

Product driven technology - seems to be the new mantra of current open source initiatives, no waste! just lean clean api's and technologies. From Netflix to Facebook.

GraphQL is absolutely no exception, it came to life five years ago to solve the version, rest endpoint madness that Facebook experienced.

So who's using this? well, you are ...

They currently have about 260 Billion requests a day! going through their GraphQL schemas with many implementations in all the major languages ( Java, Scala, JavaScript, C# ).

In this presentation I want to explain the basics and show how you can create a clean queryable, version agnostic, self documenting API that will make your consuming clients happy. Presentationen hålls på engelska

## SPRING XD

ERIK LUPANDER

Life science, transporter, råvaruutvinning, sociala medier, affärstransaktioner - enligt IBM produceras dagligen över 2.5 triljoner byte data i någon form. Hur kan man samla in, filtrera, processa och strukturerat lagra stora mängder heterogen data? Spring XD (eXtreme Data) är ett ramverk för "Data Ingestion" som syftar till att möjliggöra datainsamling i stor skala. Vi kommer titta på dess övergripande arkitektur, DSL, några exempeltillämpningar samt prata lite om hur Spring XD skiljer sig från en mer traditionell integrationsplattform.

KONFERENS  
13.00 - 18.00



## REVENGE OF THE QUIS - IN THE CLOUD

PETER LARSSON

Kommersiella molntjänster erbjuder fantastiskt kraftfulla och kostnadseffektiva applikationsplattformar (PaaS), men vad bör man tänka "efter före på" utifrån en applikations livscykelperspektiv och företrädesvis då vad gäller klassiska problemområden som interoperabilitet, portabilitet och informationssäkerhet. Vi utgår från en enkel och typisk applikation och hur denna kan hanteras med avseende på ett urval av populära och kommersiella PaaS.

## SPRING BOOT

MATS EKHAMMAR

Spring Boot erbjuder ett enkelt sätt att snabbt skapa nya Spring applikationer. Dessutom utan att behöva tänka på beroenden till 3:e parts bibliotek, initial konfiguration och paketeringen av applikationen. Vi kommer att gå igenom hur Spring Boot gör för att åstadkomma ovanstående förenkling. Därefter kommer vi att belysa några andra områden där Spring Boot väsentligt förenklar livet för en utvecklare.

## MICROSERVICES MED SPRING CLOUD OCH DOCKER: HUR GÖR MAN I PRAKTIKEN?

MAGNUS LARSSON

Många pratar om microservices och Docker som vägen fram för att bryta isär gamla tröga monoliter till mindre och mer lätthanterliga tjänster. Tjänster som sedan snabbt och enkelt ska kunna driftsättas i Docker containrar både lokalt och ute i molnet. Men hur gör man då? Att bryta isär en monolit i mindre komponenter medför att man börjar bygga distribuerade system och alla som provat på det vet vilka utmaningar det medför. Föredraget kommer handgripligen visa på hur man i en Java värld kan använda Spring Cloud och Netflix OSS för att bygga microservices som kan samverka på ett sunt sätt i en distribuerad miljö samt hur man använder Docker för att exekvera sin microservicar såväl lokalt som i molnet.

KONFERENS  
13.00 - 18.00



## JAVA 9 MODULSYSTEM (JIGSAW)

HÅKAN DAHL

Det nya modulssystemet (projekt Jigsaw) är huvudnumret på den rätt omfattande feature-listan för Java 9. Ursprungligen tänkt att ingå i Java 7 har modulsystemet nu tagit form. Modulsystemet ska råda bot på problem relaterade till classpath-röra, svag inkapsling i Java och ett monolitiskt JDK.

Vi kommer gå igenom hur modulsystemet fungerar, hur det påverkar befintliga applikationer och strategier för att använda modulsystemet.

Vi kommer även se på hur modulsystemet kan passa in i Java's ekosystem med byggsystem och repositories.

## KUBERNETES

BJÖRN BESKOW

"Divide and Conquer" är ingenjörens viktigaste sätt att angripa komplexa problem: bryt isär i mindre, begripliga delar. Objekt/komponent/service-orientering är just precis detta, tillämpat på komplexa, distribuerade mjukvaru-system. Men här finns en paradox: Vi måste kunna lita på att delarna alltid fungerar, för att helheten skall fungera. Och mjukvara är notoriskt opålitlig, hur liten den än är.

Det här kan man närra sig på två sätt: Försöka göra de små mjukvaru-delarna ofelbara (i sig en lovvärd ansträngning, men förmögligen omöjlig att lyckas med), eller helt enkelt acceptera att delar ibland slutar fungera, och göra det bästa av situationen. En microservices-arkitektur tar typiskt den andra ansatsen: Isolera delarna från varandra, och försök begränsa effekterna av att delar ibland går sönder. Container-lösningar som Docker möjliggör samma ansats: Om delar ibland går sönder, kasserar vi helt enkelt den trasiga delen och skaffar snabbt en ny (eller ännu hellre många).

Men nu ställs vi inför en ny utmaning: Hur håller man reda på väldigt många små delar, som ibland går sönder och måste ersättas, utan att det påverkar övriga delar? Open Source-projektet Kubernetes utgör Google's destillerade erfarenheter av denna utmaning. Detta föredrag belyser via en demo de centrala abstraktionerna i Kubernetes.