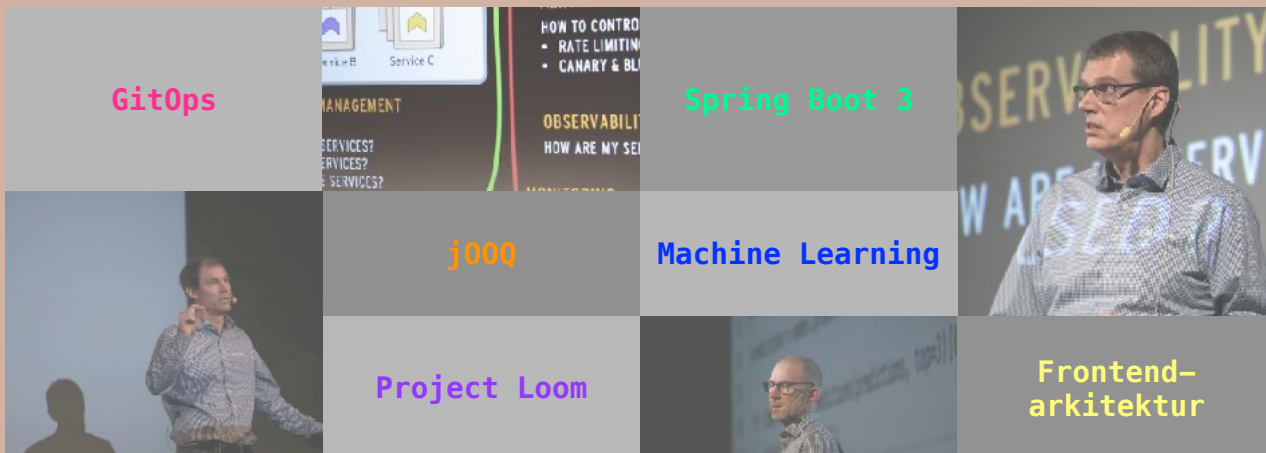


CADDEC

– CALLISTA DEVELOPER CONFERENCE –

STOCKHOLM 2023.01.19 | GÖTEBORG 2023.01.25

CADEC 2023 - KONFERENSEN FÖR UTVECKLARE SOM VILL UTVECKLAS



Cadec är konferensen som ger dig de senaste trenderna inom arkitektur och utveckling. Här får du tillfälle att rivstarta det nya året med ett rejält kunskapslyft.

Denna gång är vi tillbaka i Filmstadens biosalonger där vi kan utlova en presentationsupplevelse i toppklass.

Det blir naturligtvis också After Cadec på respektive kontor där vi kan varva ner efter konferensen med mat, dryck och mingel. En separat inbjudan till After Cadec kommer senare.

Konferensen är som vanligt kostnadsfri.

Cadec ges vid två tillfällen – ett i Stockholm och ett i Göteborg. Mer information om konferensen och länk till anmälan ser du i rutan till höger.

Antalet platser är begränsat - anmäl dig redan idag för att inte missa denna chans till gratis kompetensutveckling.

Stockholm

Datum	torsdag 19:e januari 2023
Tid	13.00 till 17.00 registrering från 12.30
Plats	Filmstaden Sergel, Hötorget 3, Stockholm
After Cadec	Mat, dryck och mingel 17.00 - Callistas kontor Drottninggatan 55, Stockholm
Anmälan	https://www.eventbrite.com/e/cadec-2023-stockholm-biljetter-459945518427

Göteborg

Datum	onsdag 25:e januari 2023
Tid	13.00 till 17.00 registrering från 12.30
Plats	Filmstaden Bergakungen, Skånegatan 16 B, Göteborg
After Cadec	Mat, dryck och mingel 17.30 - Callistas kontor Fabriksgatan 13, Gårda
Anmälan	https://www.eventbrite.com/e/cadec-2023-goteborg-biljetter-459909119557

Program Cadec 2023

Microservices för nästa årtionde med Spring Boot 3 och Spring Framework 6

Magnus Larsson

Tredje generationen av det mest populära applikationsramverket för Java, Spring Boot, är här. Eftersom det går cirka fem år mellan generationsskiftena så kan vi förvänta oss stora förbättringar. Enligt Pivotal ska Spring Boot 3 ihop med Spring Framework 6.0 lägga grunden för det kommande decenniet. Mycket fokus är lagt på att förenkla utveckling av distribuerade system, t.ex. en arkitektur baserad på mikrotjänster eller FaaS (Function as a Service).

Det här föredraget kommer belysa och demonstrera bland annat

- Hur migrerar man från Spring Boot 2?
- Kommer native-kompilering för snabbare uppstart och lägre minnesutnyttjande med GraalVM att bli vardagsmat nu?
- Observerbarhet är nu en inbyggd del i Springs ekosystem, hur kan vi utnyttja det?

Lätta trådar för tunga tillämpningar - vi synar Project Loom & Virtual Threads

Björn Beskow

Javas programmeringsmodell för trådhantering är konceptuellt enkel, men tyvärr inte särskilt lämpad för massiv skalbarhet. Trådar i JVM:en hanteras och scheduleras av det underliggande operativsystemet och är därför en dyr och begränsad resurs. Reaktiva programmeringsmodeller – som t.ex. RxJava och WebFlux – löser det problemet men på bekostnad av en betydligt mer komplex programmeringsmodell och bristande utvecklarupplevelse.

Open JDK:s Project Loom är ett försök att introducera nya, lättviktiga och därmed skalbara trådningskonstruktioner för JVM:en. Först ut från Project Loom är Virtual Threads, en programmeringsmodell som lovar att bibehålla den konceptuella enkelheten i det ursprungliga tråd-API:et – men med full skalbarhet.

I det här föredraget undersöker vi om Virtual Threads lyckas förena det bästa av två världar, och hur de passar in i Spring Boot 3.

Vill du ha världens svåraste jobb? Bli frontendarkitekt!

Stephen White

De ansvarsområden och arbetsuppgifter som frontendarkitekten behöver hantera har ökat exponentiellt i mängd och svårighetsgrad under det senaste årtiondet. Det räcker inte med att hantera traditionella funktionella och icke-funktionella krav, man behöver även adressera och hantera en ständigt växande mängd områden.

Exempel på sådana är designsystem för UX, tekniska ramverk med hög förändringstakt, komplexa mikrotjänstbaserade backends, krävande produktägare, driftsfrågor både i molnet och i olika App Stores, tillgänglighet, UI-komponenter, programmeringsspråk och ... "superstar developers". Allt detta samtidigt som resultatet ska bli en attraktiv upplevelse för användaren – och som klarar (minst) 60 fps (frames per second).

I denna presentation utforskar vi hur arkitektens traditionella roll kan tillämpas och anpassas för frontend-området. Vi berör även hur man kan tygla ökningen av entropi – inte bara genom regelverk utan också genom att kommunicera effektivt med frontendteamet som helhet för att inspirera, leda och slutligen leverera.

SQL är coolt igen - kolla in jOOQ

Jesper Holmberg

SQL är coolt igen – eller i alla fall fortfarande relevant. Relationsdatabasen, vars död förutspåddes av NoSQL-förespråkarna för tio år sedan, fortsätter att hålla en central plats i våra lösningar. De alternativ som skulle ersätta den har hittat sina nischer, men relationsdatabasen finns fortfarande kvar.

För Java-utvecklare som har arbetat mot relationsdatabaser har JPA/Hibernate länge varit det vanligaste valet. I det här föredraget tittar vi på jOOQ som ett intressant alternativ för att arbeta med SQL på JVM:en. jOOQ är ingen ny produkt – den har funnits i mer än tio år – men har utvecklats i snabb takt den senaste tiden.

jOOQ gör det möjligt att skriva SQL-nära kod i en Java-DSL på ett sätt som är lätt att arbeta med och samtidigt tillgängliggör alla de kraftfulla lösningar som modern SQL erbjuder.

Det här föredraget kommer att presentera jOOQ och visa exempel på varför man som utvecklare på JVM:en kan ha nytta av att använda verktyget – som alternativ eller komplement till de vanligare lösningarna.

Stable Diffusion - Machine Learning åt folket?

Niklas Antončić, & Marcus Björklund

Varför är det sådan hype kring Stable Diffusion? Vi förklarar och demonstrerar state-of-the-art-tekniken för generativa modeller. I april 2022 presenterades DALL-E-2 och därefter Imagen som visade sig vara banbrytande modeller inom text-till-bild-generering. Dessa modeller, av typen Latent Diffusion, var stängd källkod och ägdes av Microsoft respektive Google. I augusti 2022 släpptes dock ytterligare en variant – kallad Stable Diffusion – som öppen källkod av ett tyskt forskningsinstitut. Detta medförde en formidabel explosion av användning av Machine Learning baserad på generativa modeller.

Vi kommer att förklara lite av teorin kring modellen och demonstrera hur man kan använda den – och till och med förfina den – på en relativt modest hårdvara. Bland annat kommer du att få se exempel på text-till-bild, bild-till-bild, uppskalning och förfining.

GitOps & Pulumi - "Infrastructure as Code done right"

Andreas Tell

Moderna verktyg för Infrastructure as Code (IaC) ger möjligheten att definiera infrastruktur för olika molnleverantörer (sk "multi cloud") med välkända programmeringsspråk som TypeScript, Python, Java, Go och .NET

Förmågan att beskriva sin infrastruktur med programmeringsspråk är i sig ett stort steg framåt. Kombinerar man dessutom detta med principerna inom GitOps får man ett extremt potent helhetskoncept för att hantera infrastruktur på ett strukturerat och kostnadseffektivt sätt, med hög kvalitet och korta ledtider.

Denna presentation beskriver principerna och fördelarna med GitOps för att hantera molninfrastruktur. Vi undersöker ämnet både ur ett arkitekturellt och ett praktiskt perspektiv – det senare med hjälp av IaC-verktyget Pulumi.

Storslaget kul med småskalig ML

Pär-Anton Westbom

Idag finns det maskininlärning överallt, t.ex. när vi scrollar social media, kollar vädret, tar en bild, letar musik på Spotify etc. Ofta körs detta på stora servrar någonstans i molnet. Men i många fall vill vi ha snabb respons och har kanske inte ens tillgång till ett nätverk. Hur gör man då?

Det skall vi titta på i denna presentation där vi visar hur man kan köra maskininlärning lokalt på en liten hårdvara, en så kallad microcontroller. Historiskt sett har microcontrollers mest använts för att samla in rådata som sedan skickas vidare mer eller mindre obearbetat. Nu har processorer och minne blivit så pass små och billiga att de även kan bearbeta data lokalt och bara vidarebefordra det som är intressant.

I föredraget visar vi vad som är möjligt idag. Vi spelar in data från en microcontroller, tränar en maskininlärningsmodell på datan och lägger över den tränade modellen på microcontrollern. All databearbetning sker sedan lokalt på microcontrollern, och endast högnivåevents skickas vidare till en uppkopplad app som kan visualisera resultatet.