



# SAP Cloud Application Programming Model

<https://cap.cloud.sap>

## Roundtable #6

October 25th, 2023

PUBLIC

Today's hosts:



**Sebastian Schmidt**



**Michael  
Hellenschmidt**

# Agenda



**Introduction** (Sebastian / Michael)

05 min

**Calculated Elements** (Steffen Weinstock)

15 min

**PostgreSQL** support in CAP (Patrice Bender & Robin de Silva Jayasinghe)

30 min

**Java Native Images** with GraalVM (Marc Becker)

30 min

**Open Discussion**

10 min

# re>=CAP 2023

## Summary



re>CAP: The “(un)conference” about CAP establishing the link between the community & the development team

*Initiated by the CAP community (Volker Buzek, Gregor Wolf)*

2020



recordings

- Online via j&s soft streaming
- One track, one day
- Keynote | 4 sessions
- ~350 online attendees

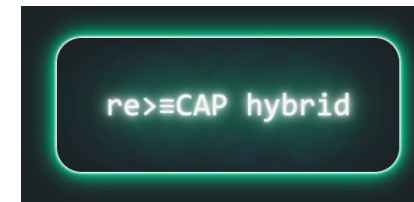
2021



recordings

- Online via global broadcast
- One track, one day
- Keynote | 10 sessions
- 940 online attendees

2022



recordings

- Speakers onsite + global broadcast
- One track, two ½ days
- Keynote | 16 sessions
- ~1000 online attendees

2023



website  
event app

- Full onsite focus (+ streaming)
- Six tracks, one day
- Keynote | Podium Discussion | 4 workshops | 12 expert sessions | 2 sponsor booths | 38 sessions
- ~1300 attendees (onsite + onsite)



6 Tracks

Keynote & Podium Discussion

38 Presentations

12 Meet-the-Experts Sessions

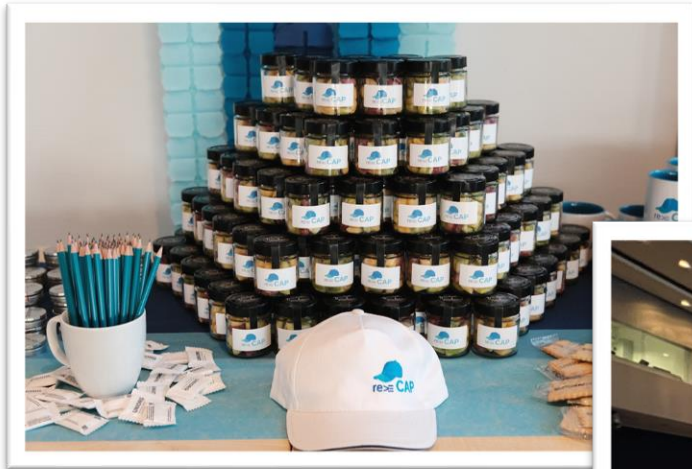
4 Hands-On Workshops

~1300 attendees


(onsite + online)




# Impressions ([link](#))



[Blog post](#) with more details



Sebastian Schmidt 

July 19, 2023 | 2 minute read

### re>CAP 2023: it's been a blast

0 15 550 Edit


[Follow](#)

[Like](#)

[RSS Feed](#)

### re>CAP 2023

Over the last week, we buzzed with excitement as our long-awaited annual reCAP developer (un-) conference unfolded. It finally took place on July 7 as the first actual onsite version of re>CAP and celebrated as a breeding ground for knowledge exchange, open-source developments, and collaboration, concluded with incredible showcases, demos and innovative ideas.



With over **50 sessions** featuring **70 speakers** across **six parallel tracks**, this event brought together developers, architects, stakeholders from around the world. Developers and tech enthusiasts from Australia, India, the United States, and many other countries came together to exchange ideas, collaborate, and foster new connections.

## Recordings

Main track: [https://broadcast.sap.com/replay/23707\\_reCAP2023](https://broadcast.sap.com/replay/23707_reCAP2023)

Side Tracks: <https://www.js-soft.com/recordings-of-recap-2023/>

# Open Sourcing CAP Webinar

Get an update on CAP's **Open Source strategy** and an **overview** about the recent Open Source additions.

Two deep-dives:

1. **PostgreSQL** support for CAP Node.js
2. **Feature Toggles** plugin

→ [Recording](#) ←





# Open Source Timeline



2019-2021	2022	2023	2024
CAP Samples: Node.js	GraphQL	CDS Typer	Odata V2-V4 converter
CAP Samples: Java		CAP Documentation	MTX Tool
CAP Samples: SFLIGHT		Audit Logging	Feature Toggle Library
		CAP Samples: Incident Mgmt.	Event Queue
		SQLite	BDD & Cucumber
		PostgreSQL	CAP Operator
		Rizing CDS Load Test	Rizing CDS Extension (CDSX)
		CDS Types*	Change Tracking
		Open Telemetry*	Attachment Service*
		Alert Notification*	SAP HANA (Early Adopter)*
		CDS Treesitter*	MS Azure Cosmos DB (Alpha)*
			Common Content*
			SAP HANA (GA)*

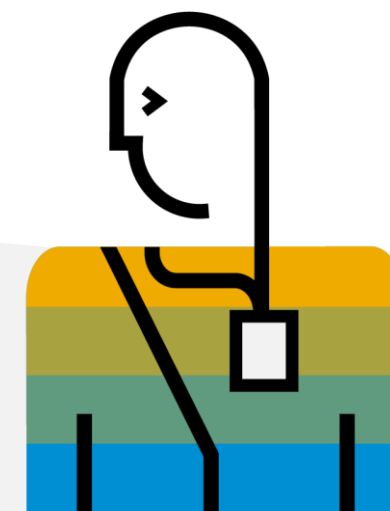
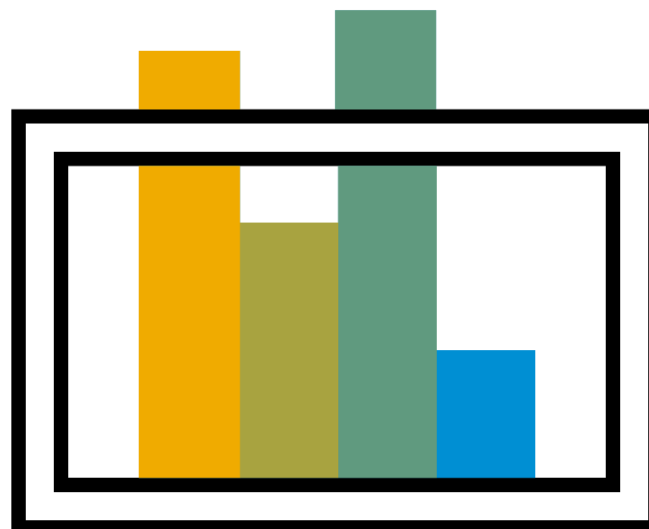
  

Samples	DB Service	Protocol Adapter	BTP Services Plugins
---------	------------	------------------	----------------------



# Presentation #1

Calculated Elements – Steffen Weinstock



# Calculated Elements



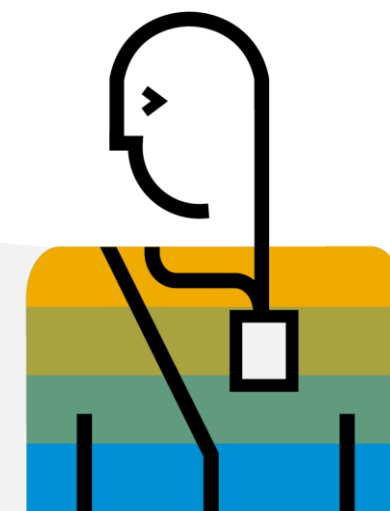
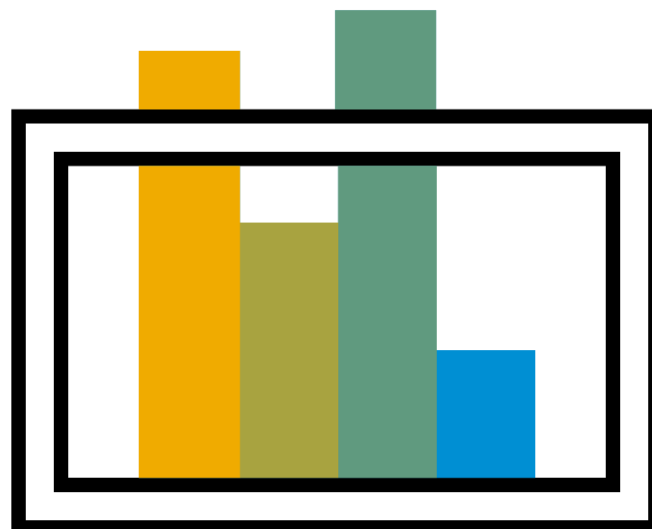
Documentation: <https://cap.cloud.sap/docs/cds/cdl#calculated-elements>

```
entity Employees {  
  key ID      : UUID;  
  firstName  : String;  
  lastName   : String;  
  name       : String = firstName || ' ' || lastName; // <--  
}
```



# Presentation #2

PostgreSQL support in CAP – Robin de Silva Jayasinghe (for Java), Patrice Bender (for Node)



# PostgreSQL Adapter: Runtime and Tool Support



- **Node.js support**

- Through new NPM package [@cap-js/postgres](#)
  - Part of new [open-source database layer](#)
- Add it to your project, e.g. with `cds add postgres`
  - Auto-wires Postgres as DB in production
- Automatic schema evolution with `cds deploy`

# PostgreSQL Adapter: Features and Community



- Benefits from **features** like

- Path **expressions & filters**:

```
SELECT `from Books { author.name as author, genre.name as genre }`  
SELECT `from Books:author[name like 'Ed%' or ID=170] { ... }`
```

- Optimized **expands** with less DB roundtrips
  - Support of **calculated elements**
  - More and portable **operators**, like `==, =, = null, <>, !=, IN, LIKE, <, >, <=, >=`
  - More and portable **functions**, like `concat, contains, search, startsWith, year, month ...`
  - Portable **functions from SAP HANA** : `years_between, ...`
  - Lean Fiori **draft**: cleaned-up impl. with cheaper DB operations
  - Consistently normalized **timestamps**
  - Overall improved **performance**

See more in the [documentation](#)

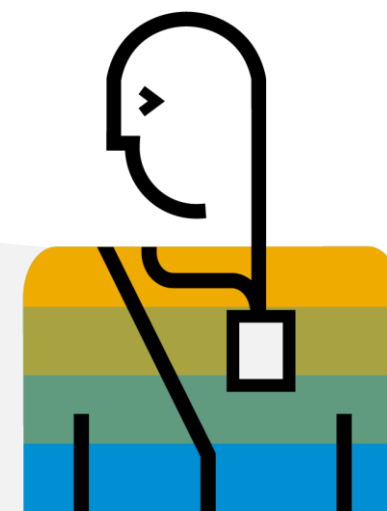
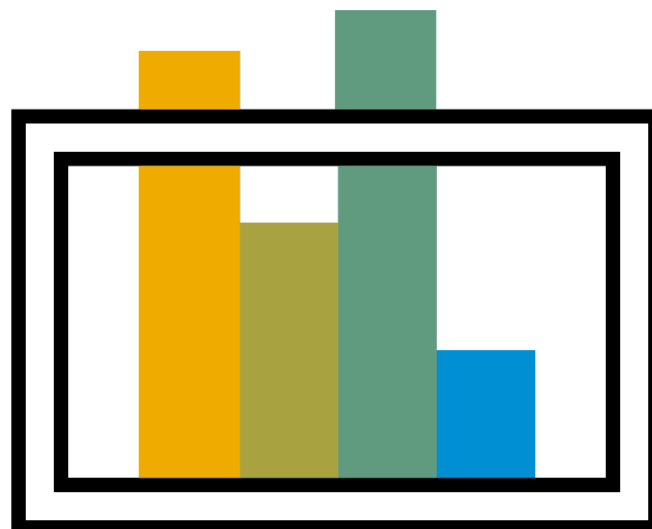
Code is open source on github.com in [cap-js/cds-dbs](#). Contributions are welcome!





# Presentation #3

Java Native Images with GraalVM – Marc Becker



# Running CAP Java Apps as GraalVM Native Images


- GraalVM Native Images: An Overview
- How do I build a CAP Java App as a Native Image?
- Possibilities & Challenges

# **CAP Java & Native Images in Action**

# GraalVM Native Images: An Overview

- Package your Java application as an ahead-of-time compiled standalone executable
- 🚀 Faster startup times (~20 times)
- 💰 Lower memory footprint (~40% less)
- Supported since CAP Java 2.0, built on top of Spring Boot 3 and Java 17
- Requires GraalVM, a special JVM distribution

# What Do Applications Need To Do?

- In most cases CAP Java apps are compatible out-of-the-box
- There are various boundary conditions
  - All code needs to be available at build time
  - Dynamic elements of your code need to be registered  `resource-`  
`config.json`
  - All Spring bean definitions are fixed already at build time

# Understanding Spring AOT Processing

- All factors effecting creation of bean definitions need to be known at build time
  - Profiles set at build time 🖱️ `cloud`
  - CAP creates bean definitions from service bindings 🖱️ `native-build-env.json`
- 💡 Native Image is built for a specific deployment target



**Let's build**

**An Idea: Scale To Zero**

# Challenges & Outlook

- Monitoring:
  - Dynatrace OneAgent [not compatible](#)
  - Open Telemetry as potential alternative
- We are very excited about this technology!
- Try out our Native Image ready samples [CAP SFlight](#) & [Java Bookshop](#)



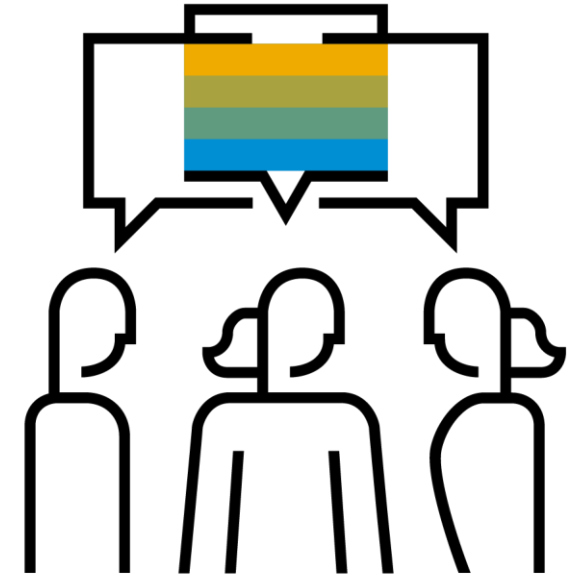
**Q&A**

# Open discussion – what's on your mind regarding CAP?



## Possible topics:

- What are you working on?
- What are you interested in learning more about?
- What feedback do you have on our products?
  - What do you like?
  - What do you need CAP to do that it does not?
  - What new capabilities do you need from CAP?
- Are you stuck in your projects?
- Would you like to demo what they you working on?





# Thank you.

Contact information:

**Sebastian Schmidt**  
Development Manager  
[se.schmidt@sap.com](mailto:se.schmidt@sap.com)

**Michael Hellenschmidt**  
Delivery Head  
[michael.hellenschmidt@sap.com](mailto:michael.hellenschmidt@sap.com)