Programmiermodelle für Unternehmensanwendungen

Cloud Application Development Kit for Scala
The SAP Innovation Center Network

2011 SAP Innovation Center founded to explore possibilities of HANA, based at Hasso Plattner Institute.

2013 Move to first building at Campus Jungfernsee (Potsdam, Germany) with around 40 employees.

2015 Establishment of the global SAP Innovation Center Network (ICN) with additional teams in Silicon Valley, Dresden, Walldorf, St. Gallen, Ra’anana, Singapore, Nanjing, Brisbane and Bangalore.

2016 Appointment of Juergen Mueller as SAP Chief Innovation Officer. (Global responsibility for driving innovation across all of SAP)

2018 The SAP Innovation Center Network is powered by more than 250 engineers, designers, product experts, and business developers.
Our Mission

Knowledge of technology & business trends

Now
Continuous innovation

Next
Adjacent innovation

New
Transformative innovation

We explore and develop meaningful new technologies to fuel transformative growth at SAP.
Projects

Knowledge of technology & business trends

Cloud Programmability
Homomorphic Encryption
Blockchain
Conversational applications

Next-gen UX
Robotic Process Automation

Knowledge of market

Now Continuous Innovation

Next Adjacent Innovation

New Transformative Innovation

Horizon 1
Horizon 2
Horizon 3
I. Scaling Development at SAP

II. Scala

III. Cloud application development kit for Scala

IV. Demo
Scaling Development at SAP
SAP - Scale

S/4HANA
The digital core

ERP with modules for finance, accounting, controlling, procurement, sales, manufacturing, plant maintenance, project system, product lifecycle management, etc.

31 languages, 61 country versions, 23 industries

on-prem and cloud

SAP HANA, ABAP, HTML5/UI5
SAP is experienced in scaling development from S/4HANA feature set.

- Leave Requests
- Freight Cost Calculation
Scaling Development

- **ABAP** is SAP’s current foundation for scalability of development
ABAP

Tailored to building (SAP) business applications (1982)

SAP NetWeaver Application Server (Development and runtime environment)

ABAP = Allgemeiner Berichtsaufarbeitungsprozessor (formerly) 
Advanced Business Application Programming (nowadays)
388,000+ Customers
91,100+ Employees
17,300+ Partners
25 Industries

€23.77B Revenue in 2017
92% of the Forbes Global 2000 are SAP Customers
#28 of Fortune's 2017 top 100 places to work
180+ Countries
SAP Cloud Platform

SAP Cloud Platform is an open platform-as-a-service (PaaS).

- Based on Cloud Foundry
- Includes
  - On-premise interoperability
  - Data storage
  - Security
  - APIs, e.g., Successfactors, Concur, S/4HANA
- Embrace variety, e.g., ABAP, Java, Node.JS, Scala

https://cloudplatform.sap.com/index.html
Application Programming Model for SAP Cloud Platform


Fiori Elements

- Approach
  - Templates
  - UI app as generic as possible
  - App-specific logic in the backend
  - Avoid low-level (hard to maintain) JavaScript
Cloud Application Development Kit for Scala

Project description

GOAL

Scale development for the SAP Cloud Platform

- Cloud application development kit for Scala
  (set of Scala libraries)

Building Enterprise Applications with a focus on

- Performance
- Concurrency Model
- Frontend / Backend in the same programming language

- Development Efficiency
- Better IDE support
- Modular building blocks

- Maintainability
- Types
Scala = Simplified + Functional Programming
(fully compatible to Java)
Why Scala?

„I think the lack of reusability comes in object-oriented languages, not functional languages. Because the problem with object-oriented languages is they’ve got all this implicit environment that they carry around with them. You wanted a banana but what you got was a gorilla holding the banana and the entire jungle.

If you have referentially transparent code, if you have pure functions — all the data comes in its input arguments and everything goes out and leave no state behind — it’s incredibly reusable.“

— Joe Armstrong, creator of Erlang
Scala – Simplicity for the JVM

```java
public class HelloJava {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

```scala
object HelloScala {
    def main(args: Array[String]): Unit = 
        println("Hello World!"))
}
```

"HelloJava.java" 5L, 122C

"HelloScala.scala" 4L, 90C
Scala – Simplicity for the JVM

```scala
object TestList {
  def test: Unit = {
    val lst = List("1", "2", "3")
    val ints = lst.map(x => x.toInt)
  }
}
```
Scala – Simplicity for the JVM

```java
public class ProductJava {
 private int id;
 private String category;

 public ProductJava(int id, String c) {
   this.id = id;
   this.category = c;
 }

 public int getId() {
   return id;
 }

 public String getCategory() {
   return category;
 }

 public boolean equals(Object o) {
   if (o instanceof ProductJava)
     return false;
   ProductJava p = (ProductJava) o;
   return
     id == p.id &&
     (category == null
     && p.category == null)
   || category.equals(p.category));
 }

 public int hashCode() {
   return 7 * id + 13 * ((category == null) ? 42 : category.hashCode());
 }
}
```

```scala
case class ProductScala(
  id : Int,
  category(): String
)```

Scala – Simplicity for the JVM

```scala
import scala.concurrent.Future
import scala.concurrent.ExecutionContext.Implicits.global

trait ScalaFuture {
  def getX: Future[Int]
  def getY: Future[Int]
  def f(x: Int, y: Int): Future[Int]

  def compute: Future[Int] =
  for {
    x <- getX
    y <- getY
    result <- f(x, y)
  } yield (result)
}
```
Scala
Feature Spectrum: Java vs. Scala

Java application developer

Java platform / library developer

Scala application developer

Scala platform / library developer

Not available in Java, C/C++, ABAP, or others

Beginner A1 A2 B1 B2 C1 C2 Guru

- OO programming
- Event loop-based
- Concurrency
- Threads
- Design patterns
- Generics
- Structural typing
- co-contravariance
- monads
- Type providers
- Type classes
- Macros
- Dependent types
Risks
Cloud Application Development Kit for Scala
Cloud Application Development Kit for Scala

Features

- Read from/write to **SAP data sources**:
  - S/4 (via OData or CDS), ERP (DDIC + DB-level access), etc.
  - Homogeneous query language for many SAP data sources (developer productivity)

- Write **Fiori/SAP UI 5 applications** using high-level abstractions **in Scala**:
  - Scala code compiles to JavaScript, using facades for UI5
  - Flexible databinding and **reactive** library for UI programming
  - Integration into **Fiori Launchpad**
Scala
Financial Accounting Example
Demo
Open Architecture

**Attract Talent**

New generation of developers

- Want to work with modern and powerful tools (Scala (2007) not Java (1995))

- Want to work in open-source environment

- Consider their github.com account as their business card
  - A tech company ideally attracts the most talented developers
  - A tech company cannot afford losing the most talented developers

- We will release the Scala project as Open Source
Summary

Scale development for the cloud age

- Development Efficiency
- Performance
- Maintainability

Attract talent
Thank you.

Contact information:

Dr. Arne Ludwig
ICN Marmolata

arne.ludwig@sap.com