Speichersparende XML Verarbeitung mit StAX und JAXB

Wolfgang Nast

Nürnberg, 19.11.2014

Agenda

- Vorstellen der Technologien
  - StAX
  - JAXB
- Gemeinsame Verwendung
- Beispiel (Demo)
- Zusammenfassung

Vorstellung der MT AG

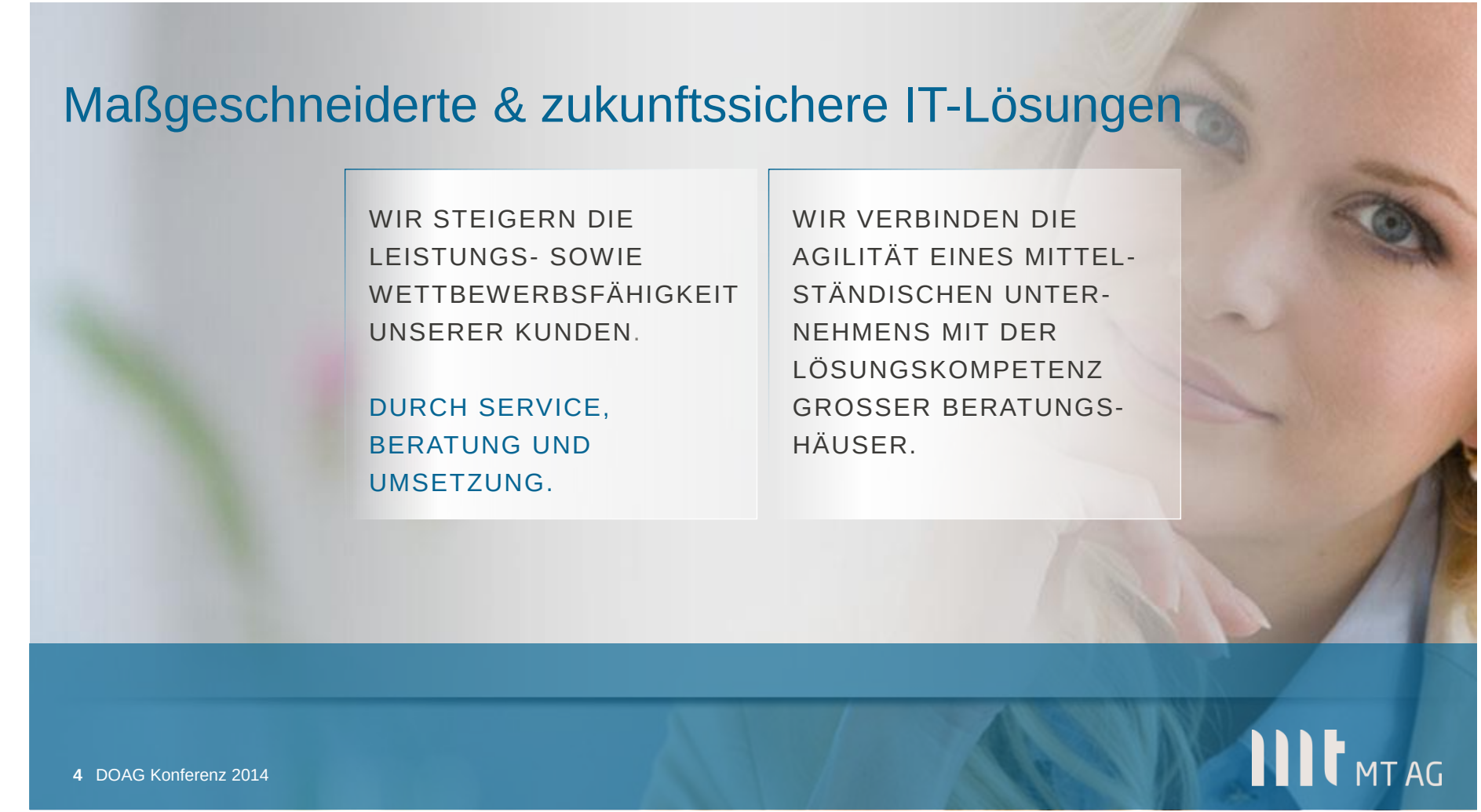
MT AG

Maßgeschneiderte & zukunftssichere IT-Lösungen

WIR STEIGERN DIE LEISTUNGS- SOWIE WETTBEWERBSFÄHIGKEIT UNSERER KUNDEN.

DURCH SERVICE, BERATUNG UND UMSETZUNG.

WIR VERBINDEN DIE AGILITÄT EINES MITTEL-STÄNDISCHEN UNTER-NEHMENS MIT DER LÖSUNGSKOMPETENZ GROSSER BERATUNGS-HÄUSER.



MT AG

FACTS & FIGURES

GESCHÄFTSFORM	INHABERGEFÜHRTE AG
HAUPTSITZ	RATINGEN
GRÜNDUNGSJAHR	1994
BESCHÄFTIGTE	180 FESTANGESTELLTE MITARBEITER
BETEILIGUNGEN	MT-IFS GMBH (RATINGEN), MT-IFS SARL (LUXEMBURG)

MT AG

UNSER PORTFOLIO

APPLICATION DEVELOPMENT

- APEX / ADF
- JAVA
- .NET

INTEGRATION SERVICES

- STRATEGIE
- ARCHITEKTUR
- SAP HANA

IT SYSTEM SERVICES

- MANAGED SERVICES
- BETRIEB
- MIGRATION

BUSINESS INTELLIGENCE SOLUTIONS

- DATA INTEGRATION
- SELF SERVICE BI
- MOBILE BI

SOCIAL BUSINESS SOLUTIONS

- COLLABORATION
- SEARCH
- SOCIAL

MOBILE SOLUTIONS

- APPS
- ABLÄUFE
- LOKALISIERUNG

MT AG

Vorstellen der Technologien

MT AG

StAX - Streaming API for XML

- objektorientierte Implementierung
- funktionale Implementierung

StAX

objektorientierte Implementierung

- Factory für die Klassen:

```
XMLInputFactory inFac = XMLInputFactory.newFactory();
XMLOutputFactory outFac = XMLOutputFactory.newFactory();
```

- Der Reader zum Einlesen:

```
XMLEventReader xreader = inFac.createXMLEventReader(in);
```

- Der Write zum Schreiben:

```
XMLEventWriter xwriter = outFac.createXMLEventWriter(out);
```

StAX

funktionale Implementierung

- Factory für die Klassen:

```
XMLInputFactory inFac = XMLInputFactory.newFactory();
XMLOutputFactory outFac = XMLOutputFactory.newFactory();
```

- Der Reader zum Einlesen:

```
XMLStreamReader xreader = inFac.createXMLStreamReader(in);
```

- Der Writer zum Schreiben:

```
XMLStreamWriter xwriter = outFac.createXMLStreamWriter(out);
```

StAX Auswahl der Implementierung

Betrachtung objekorientierte Implementierung

- Vorteile
  - XML Elemente sind Objekte
  - Startelement hat alle Namespaces und Attribute als Eigenschaften
  - aktuelles XML Element kann zurückgestellt werden
- Nachteile
  - Elemente müssen nach Abfrage des Types geholt werden
  - braucht mehr Speicher als die funktionale Implementierung
  - benutzt intern die funktionale Implementierung
  - Endelement benötigt Namespace

StAX Auswahl der Implementierung

Betrachtung funktionalen Implementierung

- Vorteile
  - die Daten werden abhängig vom Typ mit einer Methode geholt
  - holt keine Daten im Voraus(Attribute)
  - braucht weniger Speicher als objektorientierte Implementierung
  - Endelement kommt ohne Parameter aus
- Nachteile
  - Attribute müssen einzeln geholt werden
  - viele Funktionen um an die Daten abzurufen

JAXB – Java Architecture for XML Binding

- Objekte zum Lesen und Schreiben
- Generierung der Strukturen

JAXB

Objekte zum Lesen und Schreiben der Strukturen

- Der Context:

```
JAXBContext cont = JAXBContext.newInstance("package");
```

- Der Einleser für die Objekte(das XML):

```
Unmarschaller umar = cont.createUnmarschaller();
```

- Der Schreiber für die Objekte(das XML):

```
Marschaller mar = cont.createMarschaller();
```

JAXB

Generierung der Strukturen

- Das Kommando zum Generieren:

```
xjc
```

- Der Context in Java:

```
JAXBContext cont = JAXBContext.newInstance(Klasse.class);
```

- Empfehlungen:
  - Context klein halten beim Generieren
  - Context klein halten in Java

Vorteile und Nachteile von StAX und JAXB

- Stärken und Schwächen von StAX
- Stärken und Schwächen von JAXB

Stärken und Schwächen StAX

- Vorteile:
  - leichtes Navigieren zu den Elementen
  - Auslassen von Elementen
  - nur ein Element im Speicher
  - Kommentare und PI lesbar
- Nachteile:
  - Objekte müssen Elementweise gelesen warden
  - Objektstruktur wird nicht generiert
  - keine Validierung gegen das Schema

Stärken und Schwächen JAXB

- Vorteile:
  - Objektstruktur wird generiert
  - Objekt wird mit allen Unterelementen gelesen und geschrieben
  - validieren der Daten nach Schema
- Nachteile:
  - ganzes XML wird auf einmal geladen und geschrieben
  - der Context benötigt viel Speicher beim Laden und Speichern
  - keine Elemente ausgelassen
  - keine Kommentare und PI lesbar

Vereinen der Stärken und minimieren der Schwächen

MT AG

Vereinen der Stärken und minimieren der Schwächen

- Vorteile nutzen:
  - navigieren mit StAX
  - Objekte mit JAXB generieren
  - Teilvalidierung mit JAXB
- Nachteile minimieren:
  - nur Teile lesen und schreiben mit JAXB
  - Liste von Objekten mit StAX einzeln Auswählen
  - einzeln Laden mit JAXB

JAXB anpassen für Fragmentverarbeitung

Teilverarbeitung

- Ausgabe auf Fragmente Beschränken:

```
mar.setProperty(Marshaller.JAXB_FRAGMENT, TRUE);
```

- EventHandler setzen für Validierung:

```
umar.setEventHandler(this);
//Interface implementieren
public Boolean handleEvent(final ValidationEvent event) {
      LOG.info("Schema Event: " + event.getMessage();
      return true;
}
```

JAXB anpassen für Fragmentverarbeitung

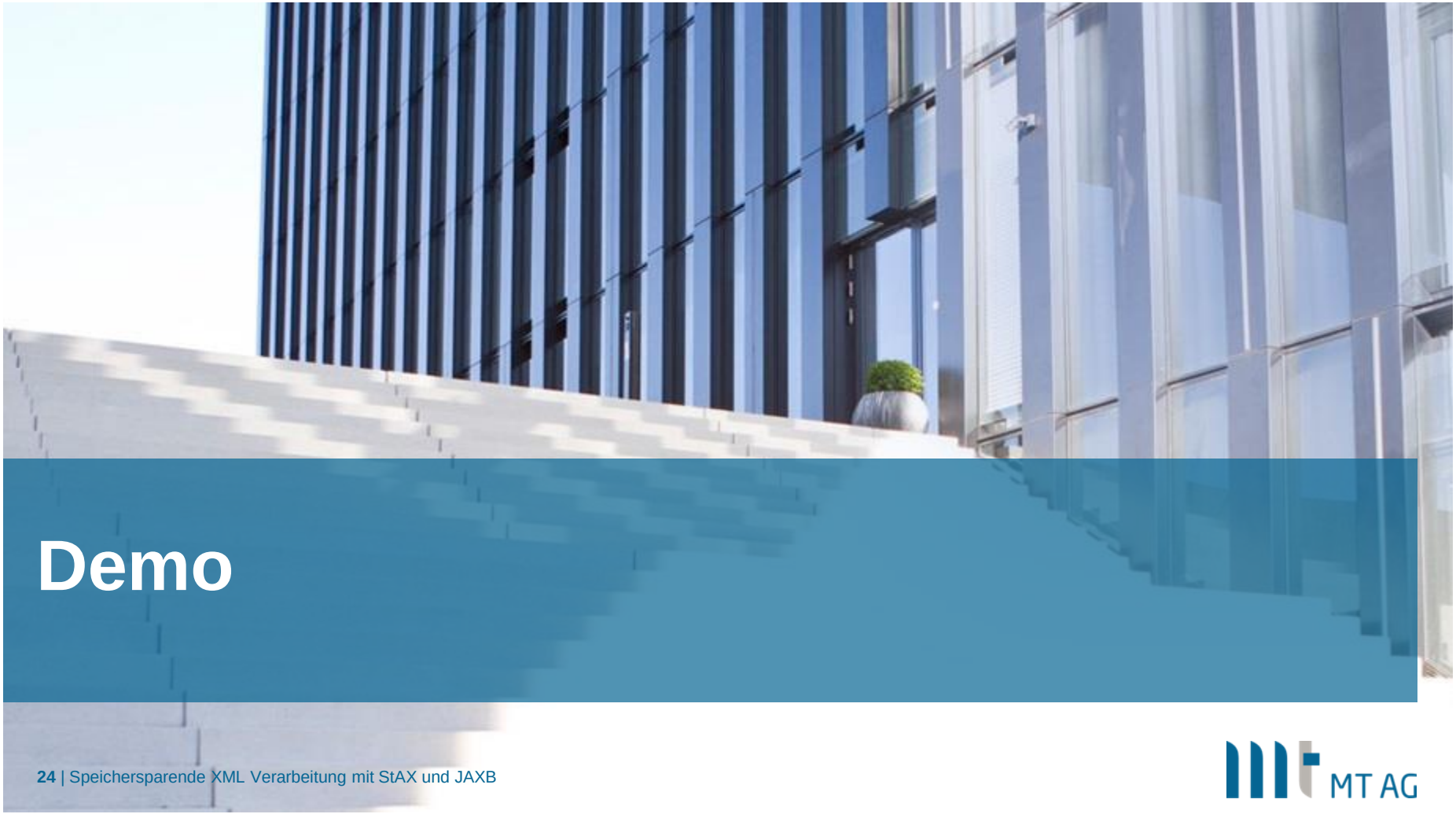
JAXBElemente erstellen

- Verpacken der Fragmente:

```
private <T> JAXBElement<T> createElem
      (final String name, T wert) {
        return new JAXBElement<>
      (new QName("namespace", name),
      (Class<T>)wert.getClass(), wert);
    };
```

Speicherverbrauch JAXB und StAX in MB

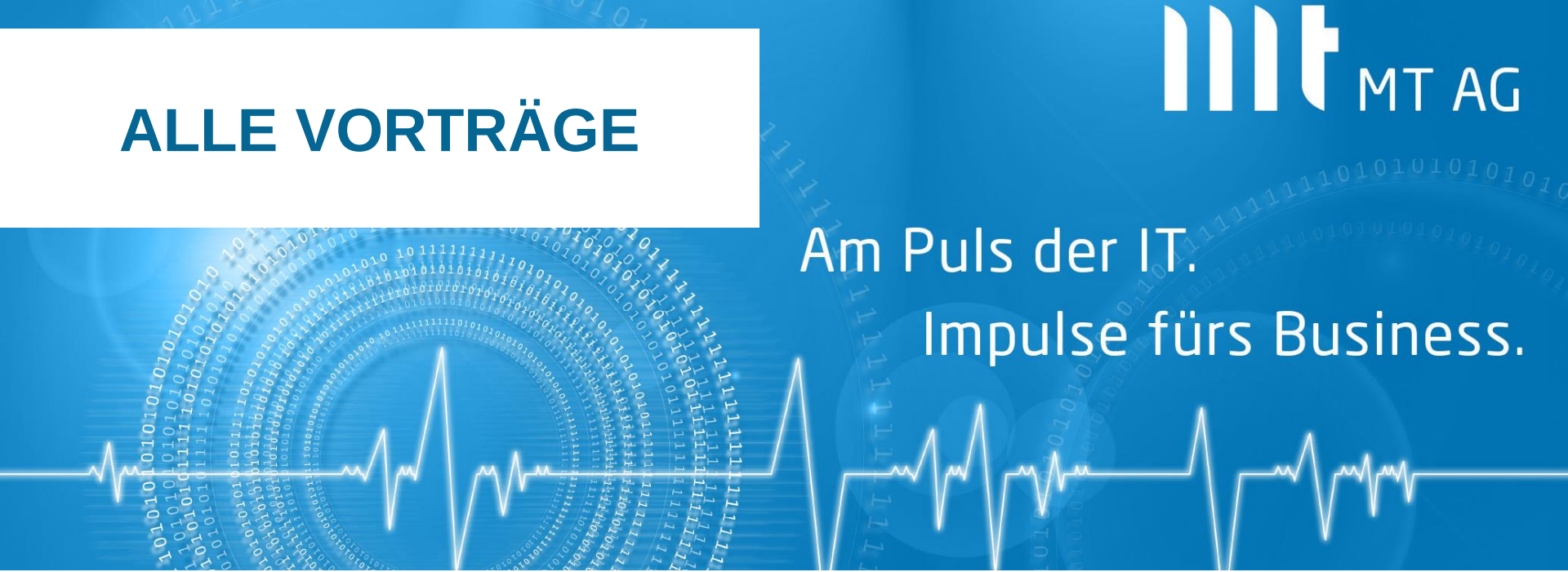
Elemente	Datei Größe	Heap JAXB	Rest JAXB	Heap StAX	Rest StAX
1	1 KB	1,0	8,8	0,7	5,5
10	1 KB	1,0	8,8	0,7	5,5
100	2 KB	1,0	8,9	0,7	5,6
1.000	12 KB	1,0	9,1	0,7	5,7
10.000	127 KB	1,4	9,5	0,7	6,0
100.000	1,4 MB	4,4	9,7	0,8	6,1
1.000.000	14 MB	34,2	9,9	0,8	6,4
10.000.000	155 MB	300,2	10,0	0,8	6,5



Demo

MT AG

ALLE VORTRÄGE



MT AG

18.11.14	14:00 - 14:45 Neu Delhi	OraSASH everybodies ASH ??	Ernst Leber
18.11.14	15:00 - 15:45 Sydney	Datenmodellierung ist langweilig, lassen Sie Datamodeler das machen	Oleg Kiriltsev
19.11.14	12:00 - 12:45 Sydney	Das nächste Duet(t): APEX und SAP.	Niels de Bruijn
19.11.14	16:00 - 16:45 Singapur	Speichersparende XML Verarbeitung mit StAX und JAXB	Wolfgang Nast
19.11.14	16:00 - 16:45 Istanbul	"Echtes" Single Sign On mit APEX realisieren.	Niels de Bruijn
20.11.14	09:00 - 09:45 Sydney	Five Fingers Death Punch	Oliver Lemm
20.11.14	10:00 - 10:45 Helsinki	12c Oracle Warehousing voll Groovy. Ein Projektbericht.	Rosenberger Ketteltasche
20.11.14	12:00 - 12:45 Istanbul	Dateien per Drag & Drop in APEX Applikationen ablegen	Franziska Höcker
20.11.14	15:00 - 15:45 Oslo	Ist Gradle auch für die APEX-Projekte?	Oleg Kiriltsev
20.11.14	15:00 - 15:45 Istanbul	Tune Up Your APEX	Oliver Lemm

MT AG

Vielen Dank.

Wolfgang Nast
Senior Consultant

Telefon: 02102 309610
Telefax: 02102 30961101

E-Mail:Wolfgang.Nast@mt-ag.com

www.mt-ag.com