FROM LEGACY TO MICROSERVICES
Lessons learned on the road to success by Miles & More

Matthias Krohn - Miles & More
Manager IT, Lead Innovation Lab

Torben Jäger - Red Hat
Specialist Solution Architect Middleware & PaaS

Serge Pagop - Red Hat
Sales Specialist Middleware & PaaS

May 2, 2017
AGENDA

➔ MILES & MORE INTRODUCTION
➔ PROJECT SCOPE AND SETUP
➔ TECHNICAL AND ARCHITECTURAL CHALLENGES
➔ COLLABORATION
➔ LESSONS LEARNED AND OUTLOOK MILES & MORE IT ARCHITECTURE
more than 20 years experience
every month 150k new members register for the Miles & More program
more than 300 Non-Air partners
every 2 minutes a brand product is shipped
in 1 hour 5.9 million Air-Miles get earned – 84% will be redeemed
more than 1.4 million Miles & More creditcards

EXPERTS FOR CUSTOMER LOYALTY – WORLDWIDE
Miles & More Introduction
FREQUENT FLYER PROGRAM TO MULTI-PARTNER-PROGRAM
Carve Out Miles & More from Lufthansa in 2015
IT FACING CHALLENGES AFTER MILES & MORE CARVE OUT

Miles & More high-level IT architecture

- Many technical and operational interfaces
- Heterogeneous platform
- Heavy-duty and complex systems
IT FACING CHALLENGES AFTER MILES & MORE CARVE OUT

Miles & More 2-Speed-IT approach

1. **LOW RISK – MAXIMISE CONTINUITY**
   - Low risk
   - No support for new strategy
   - Long time-to-market
   - Hard to maintain and improve

2. **MAXIMISE COMPLEXITY REDUCTION**
   - High risk
   - Huge investment for rebuild whole infrastructure
   - Remaining high complexity
   - Timeline > 2 years

3. **GUARANTEE CONTINUITY & MAXIMISE AGILITY**
   - Manage complex systems and guarantee performance & stability
   - (Re-)build systems in micro services, ensure a suitable time-to-market and support for innovations
   - Encapsule main functions from complex systems in micro services

"Do nothing"

"Everything from scratch"

"2-Speed-IT"
NON-AIR PARTNER INTERFACE: LEGACY TO MICRO SERVICES
Project Scope and Setup

Enterprise Service Bus
- SFTP
- HTTPS

DB-Cluster
- Business Rules
- Validation Rules
- Mail Service
- Cronjobs
- Batch Processing

Non-Air Core Application
- Database
- Persistence Service
- Integration Service
- Maintenance Service

DWH
- Report Service
- Search Service

Search Engine
- Access Control
- Partner Administration
- User Administration

Non-Air Core IDM Modul
- Business Rules
- Validation Rules
- Mail Service
- Persistence Service
- Maintenance Service
- ZR6 / SAMBA Connector
- GUI
- Report Service
- Search Service
- Integration Service

Access Control
- GUI
- SFTP
- HTTPS
- SSH

Validation Rules
- Business Rules
- Export Service
- Mail Service
- Cronjobs

Persistence Service
- Database
- Cronjobs
- Maintenance Service

Search Service
- Search Engine
more than 90% of our non-air partners are already migrated

50 involved project member and stakeholder

approx. 50 implementation project schedule 8 month

more than 50 micro services

less than 1% above budget

25 virtual server
TECHNICAL AND ARCHITECTURAL CHALLENGES
HISTORY
Technical and Architectural Challenges

T0: Support Tickets
T1: Debugging Sessions
T2: Analysis Action Items
T3: Results Workshop
OPENSHEET SYSTEM ARCHITECTURE
Technical and Architectural Challenges

1
Master Infra Worker

1
Master Infra Worker

1
Master Infra Worker
**JUST A FEW NUMBERS**  
Technical and Architectural Challenges

<table>
<thead>
<tr>
<th>vCPUs</th>
<th>GB of RAM</th>
<th>Microservices</th>
<th>A-MQ</th>
<th>Infrastructure</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>8</td>
<td>21</td>
<td>2</td>
<td>10</td>
</tr>
</tbody>
</table>
OPENSSHIFT SYSTEM ARCHITECTURE - NEXT GEN
Technical and Architectural Challenges

Master
Infra
Worker
LESSONS LEARNED
Technical and Architectural Challenges

# automation
# separation of concerns
LESSONS LEARNED
Technical and Architectural Challenges

# monitoring
# LESSONS LEARNED

Technical and Architectural Challenges

# timing
LESSONS LEARNED
Technical and Architectural Challenges

# education
RECOMMENDED WAY TO START Collaboration

Phase 1: Smart Start
- Discover
- Design
- Container Platform Design Workshop

Phase 2: Groundwork
- Discover Design Deploy
- Operationalizing Container Platform
- Container Driven Continuous Delivery

Phase 3: Automation Rollout
- Discover Design Deploy
- Container Adoption for Application Development
- Migrating Workloads to Containers
- Microservices Development

Transparency, Visibility, Monitoring & Optimization
LESSONS LEARNED AND OUTLOOK
MILES & MORE IT ARCHITECTURE
LESSONS LEARNED

- Micro service ≠ easy
  → The total of all micro services = still complex but easier manageable

- You need to know …
  - … what you want to do before you get started
  - … how you want to do it

→ We recommend pre-project planning

- Your implementation partner should have relevant knowledge about Openshift and micro services prior to the project → We lost 4 weeks because our implementation partner acquired knowledge in first project phase

- Don’t forget your staff – internal knowledge about Openshift is essential → Similar to our implementation partner we acquired knowledge during the project and not prior to it

- Red Hat consulting and technical account management is recommended early in the project for …
  - … system fine tuning and architecture review (sizing, best practices …)
  - … support in automation (ansible scripts) and IT operations …

→ We contracted Red Hat in the middle of the project and might have avoided 4 weeks delay
HIGHLEVEL MILES & MORE TARGET IT ARCHITECTURE

(Partner) Channels

Legacy Systeme

Content Management
Adobe Analytics
Adobe
Campaign
Digital Experience (Adobe AEM)
Adobe Analytics
Content Management

Integration Bus
KDI Services
MCE
APP Services
Payment Services
WorldShop Discover

Rules Engine

SSO

API Gateway

ascale

redhat

redhat

redhat

redhat

redhat

redhat

redhat

redhat

redhat

redhat

redhat

redhat

Jboss

Fusen

redhat

ANSIBLE

Legacy Systeme

#redhat #rhsummit
THANK YOU

plus.google.com/+RedHat
linkedin.com/company/red-hat
youtube.com/user/RedHatVideos
facebook.com/redhatinc
twitter.com/RedHatNews
LEARN. NETWORK. EXPERIENCE OPEN SOURCE.