OPITZ CONSULTING

Menschen. Innovationen. Lösungen.

„Orchestrator“

IT-Paradigmenwechsel im Zeitalter des Cloud Computing

Mohammad Esad-Djou, Solution Architect
OPITZ CONSULTING Deutschland GmbH

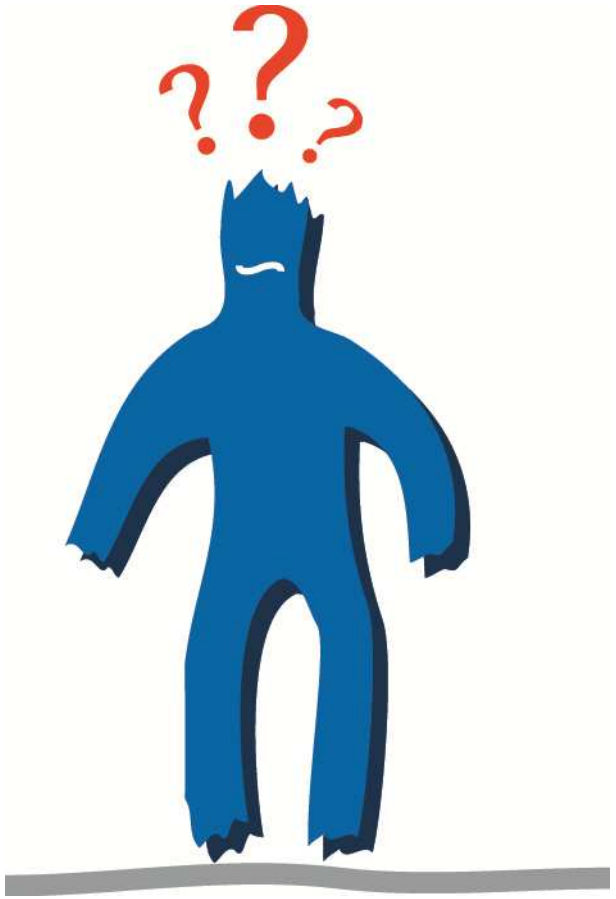
Mainz, 06. Juni 2013, DOAG 2013 IM Community Summit

Agenda

- **Problemstellung**
- **Erste Dimension: Horizontale Komplexität**
- **Zweite Dimension: Vertikale Komplexität**
- **Dritte Dimension: IT-Strategie**
- **Vierte Dimension: Plattformübergreifende Probleme**
- **Was nun? Rahmenarchitektur für eine Lösung: Orchestration**
- **Zusammenfassung**

IT-Landschaft wird noch komplexer!

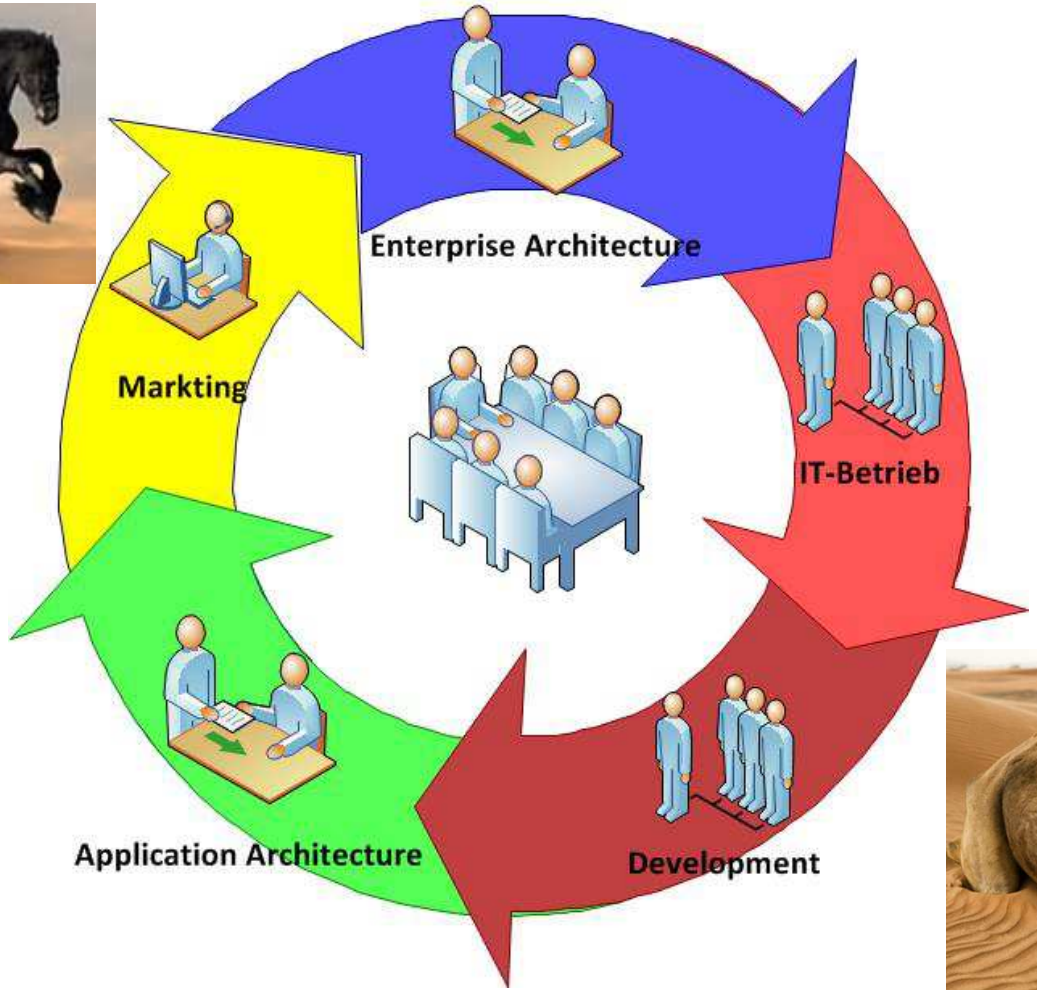
- **die klassische Kluft zwischen den Abteilungen**
- **versteckte Komplexität von Hardware- und Software-Welt**
- **Neue IT-Strategien und Entwicklungen**
- **Cloud Computing als verteiltes Echtzeit-System**



Erste Dimension: Horizontale Komplexität

- **Beziehungen zwischen den Abteilungen in Großunternehmen**
  - **EA und App. Aechitektur-Abteilungen**
  - **Development**
  - **IT-Betrieb**
  - **Marketing**
- **“DevOps”: Worum es geht?**
- **Welche DevOps Lösungen vorhanden sind?**
- **Wie kann man von Best-Practice-Pattern nutzen?**
- **Vorteile und Nachteile**

A camel is a horse designed by a committee!



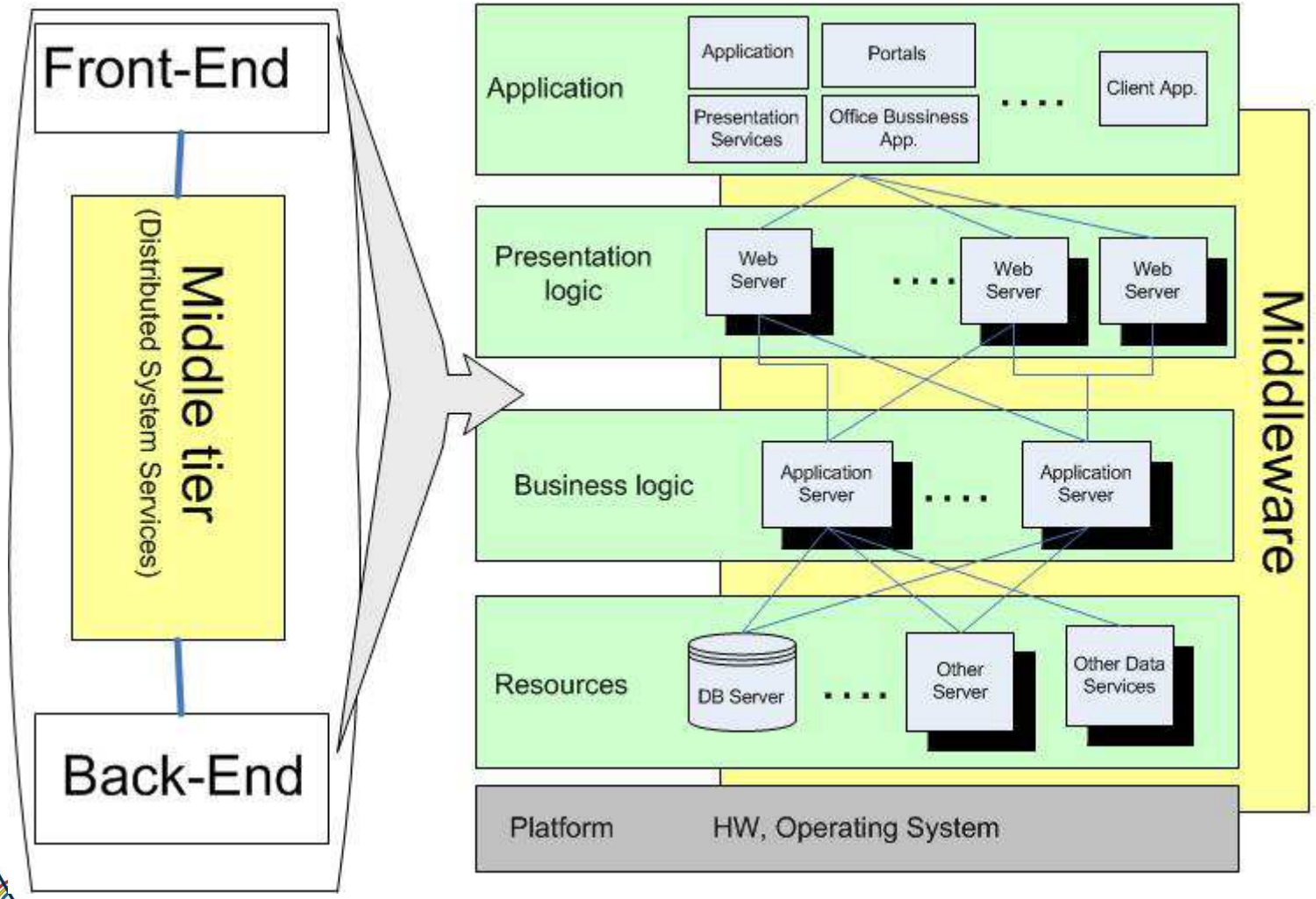
DevOps: Vorteile

- **Automatisierung**
- **Aufmerksamkeit verlagert sich auf die Kluft zwischen den Abteilungen, besonders Entwicklung und Betrieb**
- **Betonung auf Änderung von Arbeitskultur**
- **Entdeckung eine Lücke: interdisziplinäre IT-Experten**

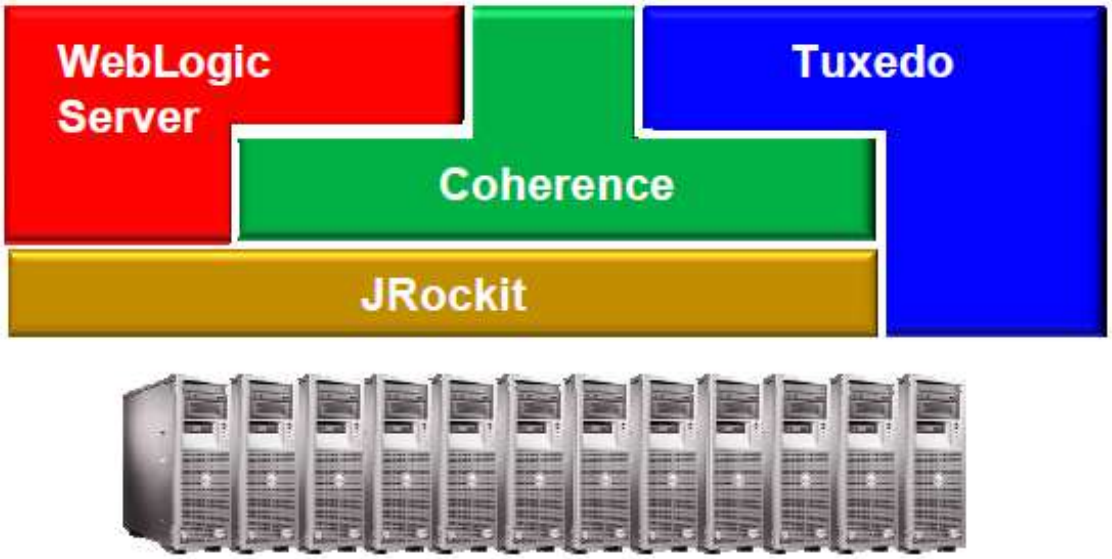
DevOps: Nachteile

- **Die Aufmerksamkeit verlagert sich NUR auf die Kluft zwischen den Abteilungen?**
- **Es fehlt ein klares Gesamt-Konzept**
- **ohne logische Konsequenz**
  - **Teambildungsmaßnahmen**
  - **Weiterbildungsmaßnahmen**
- **Last but not least: Wie wird unternehmenserfolg gemessen?**

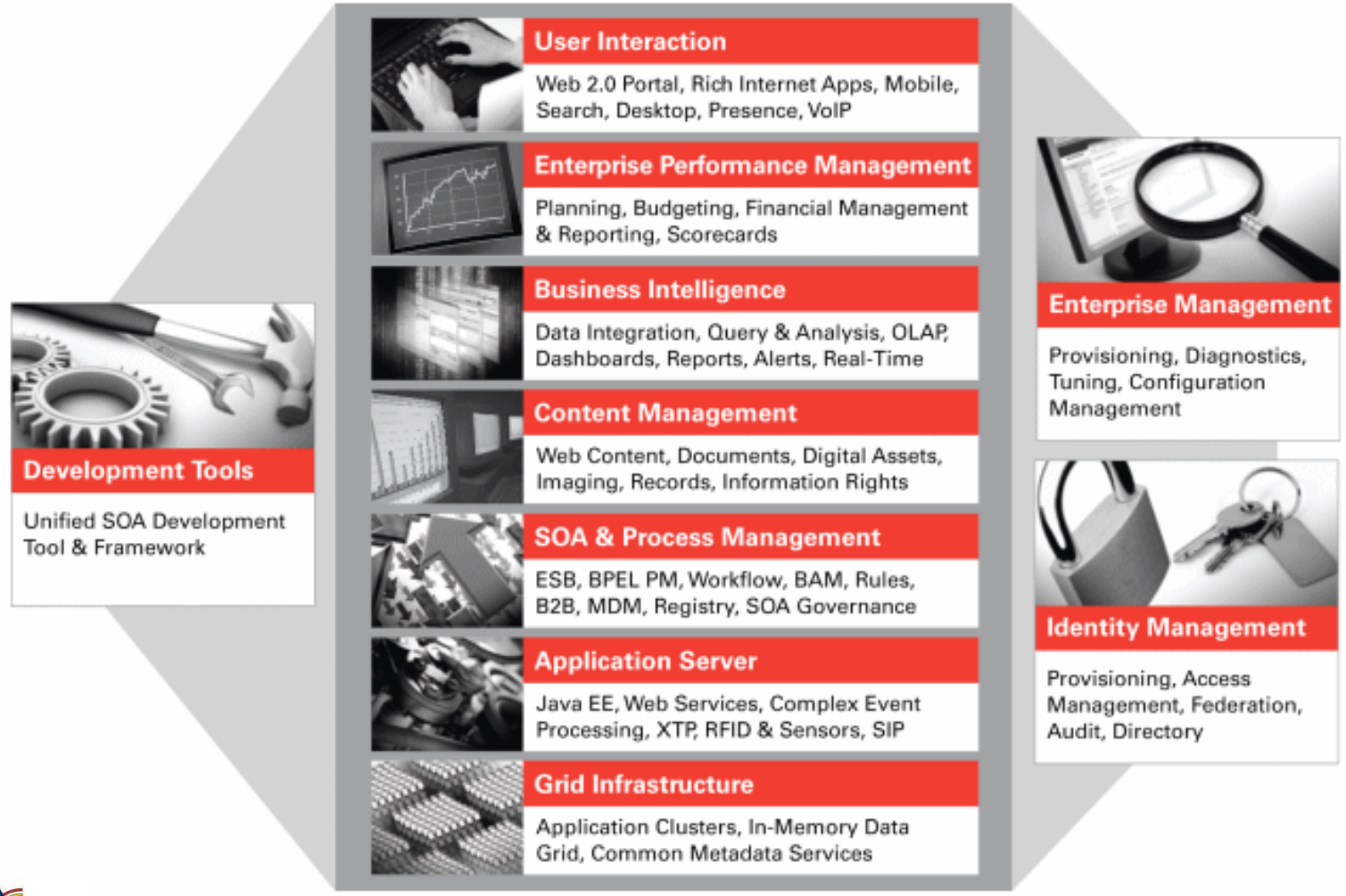
Vertikale Komplexität: Schichtenarchitektur



Beispiel: Application Server

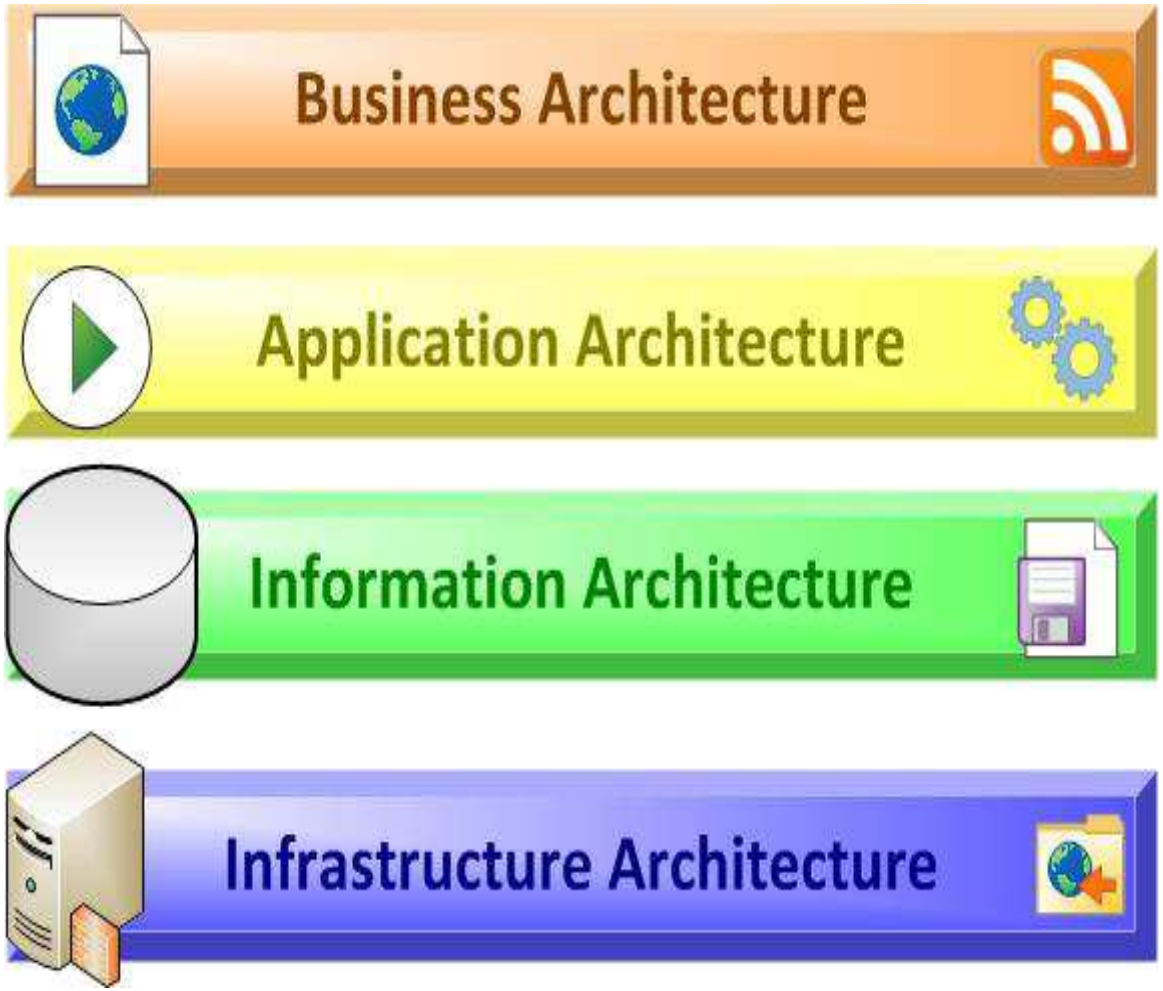


Beispiel: Oracle Fusion Middleware



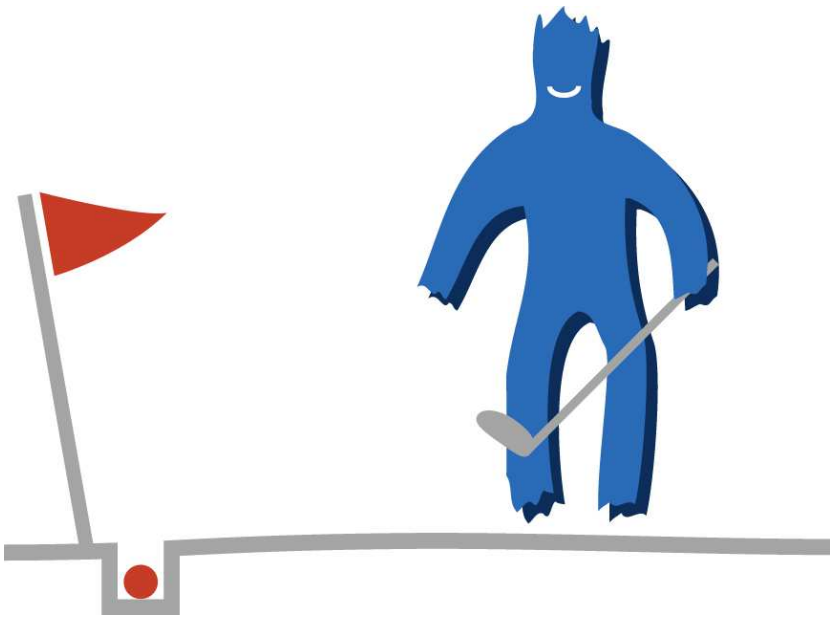
Vertikale Komplexität: Schichtenarchitektur

- **Wartbarkeit**
- **Sicherheit**
- **Schwachstellen**
- **Automatisierte bzw. Semi-Automatisierte Upgrade, Deployment**
- **Schichtenübergreifende Probleme**



Dritte Dimension: IT-Strategie

- **Diskrepanz zwischen strategischer, taktischer und operativer Sicht**
- **Enterprise Architecture (EA)**
- **IT-Architektur**
- **IT-Projekte**
- **IT-Governance**
- **IT-Prozesse und Personal**



SOA and EA Architecture Domains

Architecture domains	Software tiers (figure 4)	SOA	EA
Business	A part of Business logic; Presentation logic and Application → "Process & Rule"	Business process	Business architecture
Applications	A part of Business logic	Services and components	Application architecture
Integration	A part of Business logic	Integration architecture / ESB	Technology architecture
Data	Resources	Data architecture	Information architecture
Operations	Platform	QoS, security, monitoring, and infrastructure	Technology architecture

SOA and EA: gemeinsame Zuständigkeit

Topic	EA	SOA
similar architectural domains	✓	✓
closely align IT with business	✓	✓
use input based on business objectives	✓	✓
require similar strategies and planning activities	✓	✓

http://modj.org/home/aktueles/enterprise-architecture-ea-and-middleware/31c2030dc9cfa9730a3f85c888322f24.html

EA Domain

Topic	EA	SOA
macro level	✓	☒
business components	✓	☒
application frameworks and enterprise applications	✓	☒
enterprise-level infrastructure including servers, DB, etc.	✓	☒
All integration patterns and when they should be used	✓	☒

http://modj.org/home/aktueles/enterprise-architecture-ea-and-middleware/31c2030dc9cfa9730a3f85c888322f24.html

SOA Domain

Topic	EA	SOA
micro level	☒	✓
business services	☒	✓
service modelling	☒	✓
infrastructure that supports services, namely the Enterprise Service Bus	☒	✓
Only integration approach based on using services	☒	✓

http://modj.org/home/aktueles/enterprise-architecture-ea-and-middleware/31c2030dc9cfa9730a3f85c888322f24.html

Architektur-Probleme

Potential problem	Solution
With all the focus on SOA, other EA aspects are ignored	reciprocally cooperation between both departments (EA and SOA) for a common architecture on enterprise level → Technical Cooperation Centre
Inefficiencies as a result of duplicated efforts and missed opportunities to leverage existing architecture artefacts	Optimizing monitoring, cooperation and integration between EA, MW, and also SOA → using: Enterprise Service Bus (ESB) as integration tool
Failure to identify and incorporate SOA-specific needs as part of EA	- Separation between SOA and non-SOA projects - SOA as the basis for the EA functional architecture domain

http://modj.org/home/aktueles/enterprise-architecture-ea-and-middleware/31c2030dc9cfa9730a3f85c888322f24.html

Governance-Probleme

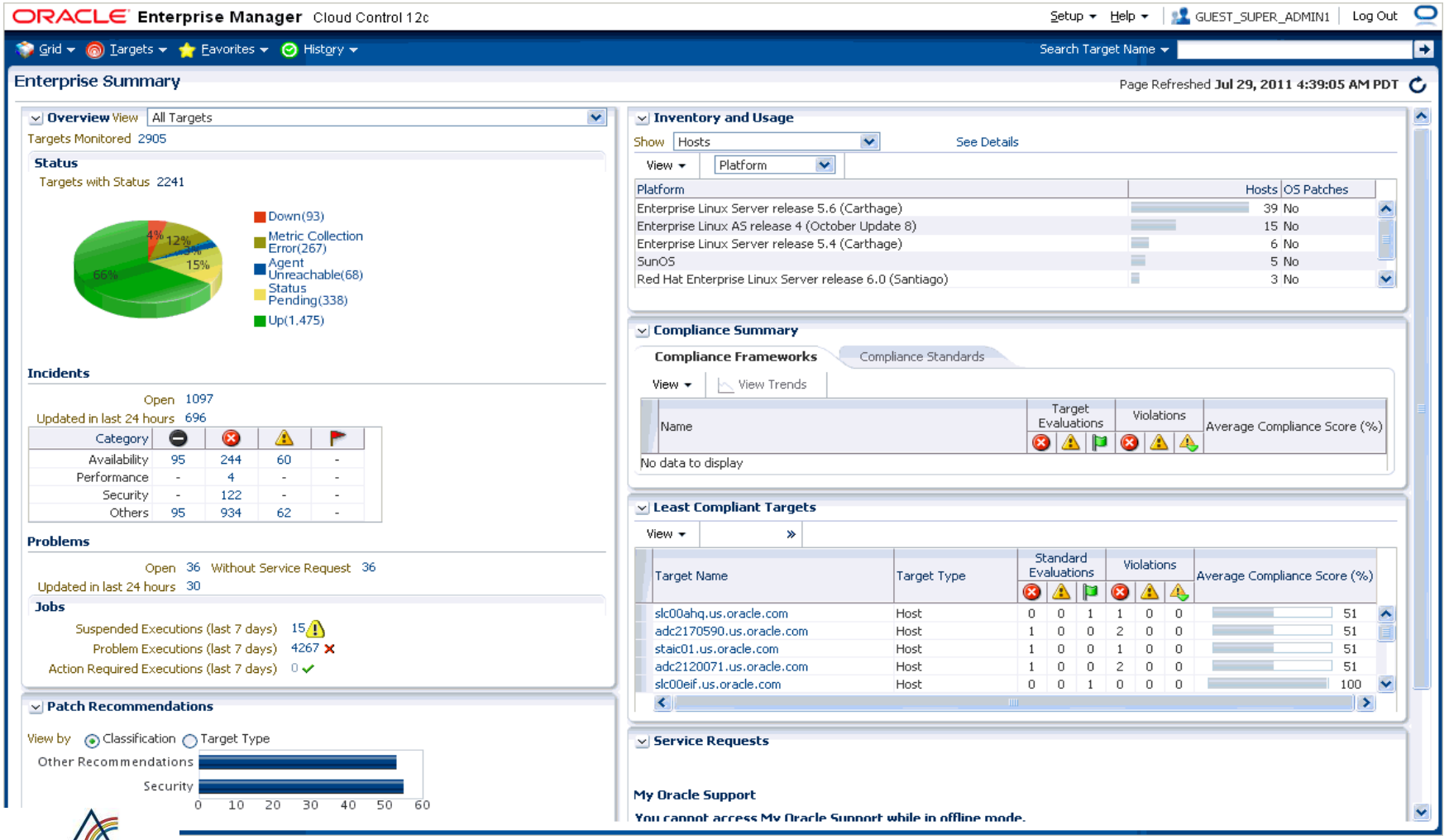
Potential problem	Solution
Overlap between the responsibilities of the SOA lead and the enterprise architect	Build-up the “Technical Cooperation Centre” help to “common decision-making”
Competition between SOA and EA for the same business resources	Sharing the same governance boards for both SOA and EA allowed business resources to address both SOA and EA needs in the same forum.
Potential for making contradicting architectural decisions that affect the whole enterprise	Except specific SOA-related issues, all other architectural decisions were approved by the EA

http://modj.org/home/aktueles/enterprise-architecture-ea-and-middleware/31c2030dc9cfa9730a3f85c888322f24.html

Was nun? Rahmenarchitektur für eine Lösung

- **Orchestration: Definition**
- **Orchestrator vs. DevOps**
- **Technologien**
  - **Oracle-Ansatz**
  - **Microsoft-Ansatz**
  - **Weitere Ansätze**
- **IT-Experten als „Orchestrators“: Voraussetzung**

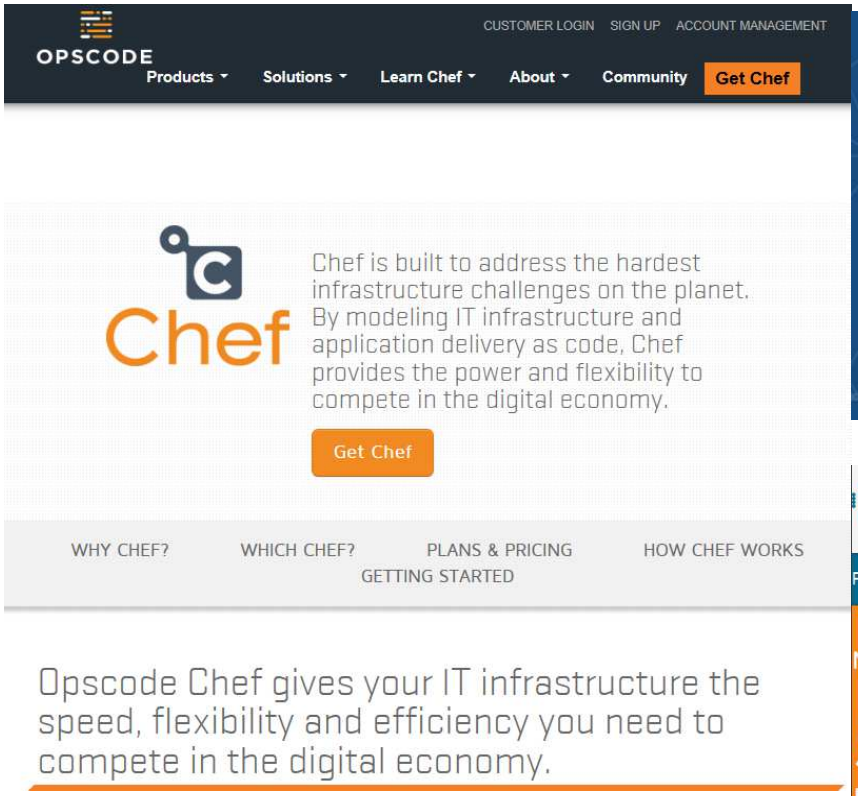
Werkzeuge: Oracle CloudControl



Werkzeuge: Microsoft Orchestrator

Workflow management solution für Datacenter

Werkzeuge: Konfigurationsmanagement

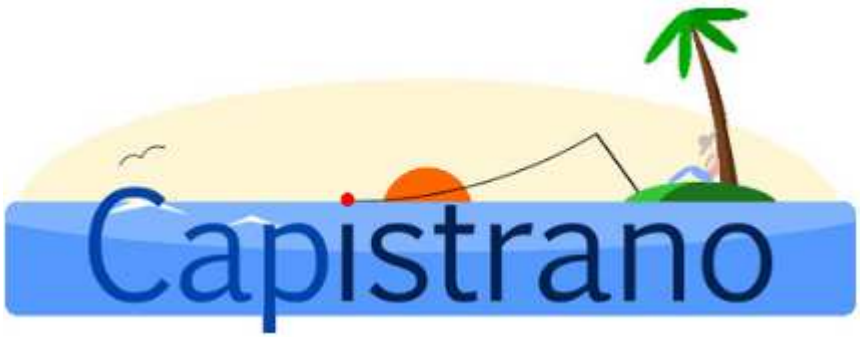


Chef

Puppet

CFEngine

Werkzeuge: Automatisierung von Deployment



Fabric

About

Fabric is a Python (2.5 or higher) library and command-line tool for streamlining the use of SSH for application deployment or systems administration tasks.

It provides a basic suite of operations for executing local or remote shell commands (normally or via `sudo`) and uploading/downloading files, as well as auxiliary functionality such as prompting the running user for input, or aborting execution.

Typical use involves creating a Python module containing one or more functions, then executing them via the `fab` command-line tool. Below is a small but complete "fabfile" containing a single task:

```
from fabric.api import run

def host_type():
    run('uname -s')
```

Once a task is defined, it may be run on one or more servers, like so:

```
$ fab -H localhost,linuxbox host_type
[localhost] run: uname -s
[localhost] out: Darwin
[linuxbox] run: uname -s
[linuxbox] out: Linux

Done.
Disconnecting from localhost... done.
Disconnecting from linuxbox... done.
```

Werkzeuge: Automatisierung von Testen und Monitoring

Cucumber *behaviour driven development with elegance and joy*

1: Describe behaviour in plain text

```
Feature: Addition
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers

  Scenario: Add two numbers
    Given I have entered 50 into the calculator
    And I have entered 70 into the calculator
    When I press add
    Then the result should be 120 on the screen
```

2: Write a step definition in Ruby

```
Given /I have entered (.*) into the calculator/ do |n|
  calculator = Calculator.new
  calculator.push(n.to_i)
end
```

3: Run and watch it fail

```
$ cucumber features/addition.feature
Feature: Addition  # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
  Scenario: Add two numbers                    # features/addi
    Given I have entered 50 into the calculator  # features/step_
      uninitialized constant Calculator (NameError)
      ./features/step_definitons/calculator_steps.rb:2:in `Given
      features/addition.feature:7:in `Given I have entered 50 into
    And I have entered 70 into the calculator    # features/step_
    When I press add                             # features/addi
    Then the result should be 120 on the screen  # features/addi
```

4. Write code to make the step pass

```
class Calculator
  def push(n)
    @args ||= []
    @args << n
  end
end
```

5. Run again and see the step pass

```
$ cucumber features/addition.feature
Feature: Addition  # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
  Scenario: Add two numbers                    # features/addi
    Given I have entered 50 into the calculator  # features/step_
    And I have entered 70 into the calculator    # features/step_
    When I press add                             # features/addi
    Then the result should be 120 on the screen  # features/addi
```

6. Repeat 2-5 until green like a cuke

```
$ cucumber features/addition.feature
Feature: Addition  # features/addition.feature
  In order to avoid silly mistakes
  As a math idiot
  I want to be told the sum of two numbers
  Scenario: Add two numbers
    Given I have entered 50 into
    And I have entered 70 into t
    When I press add
    Then the result should be 12
```

Jenkins

An extendable open source continuous integration server

cucumber-nagios

IT-Experten als „Orchestrators“

- Auf der untersten Ebene eine Orchestrierung ist ein Mensch, ein IT-Experte.
- Interdisziplinäre IT-Experten auf strategische, taktische und operative Ebenen
- Generalisten vs. klassischen Spezialisten?

Business Skills for IT Professionals

Build "T-Shaped"—Business and Technical—Skills with WebSphere Education

It’s no longer enough for a programmer to be just a programmer; in today’s multi-function world, even techies need to possess more of the skills traditionally associated with other business functions—skills like strategic thinking, interpersonal communication, project management and business analysis. IBM calls this new breed of technology skills ‘T-Shaped.’ Like the capital letter "T", the ‘T-shaped’ skilled professional has both broad and deep capabilities, encompassing both business skills and deep technical understanding.

Zusammenfassung

- IT-Landschaft wird noch komplexer!
- Mehr dimensionale Komplexität von IT-Landschaft
- Wir können flexibler mit den Problemen und Herausforderungen umgehen
- Mehrdimensionale Denkmuster
- Aktualisierung von Enterprise Architecture
- Neue Maßnahmen für Team- und Weiterbildung
- „Orchestrator“ ist daher mehr als nur ein klassischer Administrator. Er ist eine neue Generation von Experten

References for further Information

- Middleware and oracle fusion approach 11g: http://modj.org/home.html

http://modj.org/index.php?id=3&tx_ttnews[tt_news]=7&cHash=f953c555a27bc2b6274cdfe214ba3fd1

- Middleware: a short classification http://modj.org/index.php?id=3&tx_ttnews[tt_news]=6&cHash=2405bf47d8fb5efe87e291deb6135b93
- http://www.somic.org/2010/03/02/the-rise-of-devops/
- Getting Started with System Center 2012 – Orchestrator: http://technet.microsoft.com/en-us/library/hh420344.aspx
- Orchestration (computing): http://en.wikipedia.org/wiki/Orchestration_(computing)
- Onisick, Joe, "Private Cloud Automation, Orchestration, And Measured Service," Network Computing, June 23, 2011.: http://www.networkcomputing.com/private-cloud-tech-center/231000293
- Juju: https://juju.ubuntu.com/docs/about.html
- Oracle BPEL Process Manager: http://www.oracle.com/technetwork/middleware/bpel/overview/index.html
- WebSphere software: http://www-01.ibm.com/software/websphere/

Fragen?

Danke

für Ihre Aufmerksamkeit!

Mohammad Esad-Djou, Solution Architect

OPITZ CONSULTING Deutschland GmbH
Mohammad.Esad-Djou@opitz-consulting.com
Fon +49 89 680098-1409
Mobil: +49 173 7279576