



DOAG Konferenz 2011

Automatisiertes Konfigurationsmanagement mit Puppet

Daniel Schulz

Solution Architect

OPITZ CONSULTING Gummersbach GmbH

Nürnberg, 15.11.2011



Ihr ORACLE Center of Excellence

Leistungsangebot

- Java
- SOA/BPM
- ORACLE
- BI/DWH
- Outtasking
- Exadata
- Exalogic

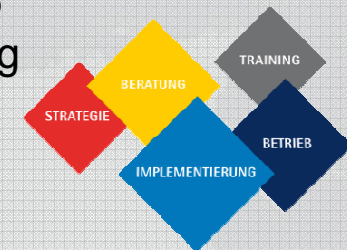


Kunden / Kollegen

- Über 600 Kunden
- Branchenübergreifend
- Über 450 Kollegen
- An 8 Standorten

Tätigkeitsfelder

- IT-Strategie
- Beratung
- Implementierung
- Betrieb
- Training



Specialized
Oracle Database



Specialized
Service-Oriented Architecture



Specialized
Oracle Enterprise Linux



Specialized
Oracle Real Application
Clusters



Specialized
Oracle Business Intelligence
Foundation

Besuchen Sie auch die anderen Vorträge von OPITZ CONSULTING und unseren Stand (Nr. 236)!

Dienstag, 15. November 2011	Mittwoch, 16. November 2011	Donnerstag, 17. November 2011
MySQL in an Oracle driven datacenter 10:00 bis 10:45 Uhr, Raum Singapur	Das ungleiche Paar – Koexistenz von OWB und ODI 09:00 bis 09:45 Uhr, Raum Kopenhagen	Grails – Die Suche ist vorbei 09:00 bis 09:45 Uhr, Raum Riga
Oracle Forms meets BI 10:00 bis 10:45 Uhr, Raum Kiew	Praxis Knowhow: Skalierung von SOA Suite 11g Cluster 09:00 bis 09:45 Uhr, Raum Budapest	Enterprise Architecture Deliverables – Let's talk about results! 09:00 bis 09:45 Uhr, Raum Prag
Minimale Latenz – Bedarfsgerechte Bereitstellung von Daten im DWH 10:00 bis 10:45 Uhr, Raum Kopenhagen	RAC ONE Node 11.2.0.2. – Wo ist meine Instanz? 13:00 bis 13:45 Uhr, Raum St. Petersburg	Brückentechnologie – Min. Downtime Plattform-Migration / Upgrade von 9 nach 11.2 10:00 bis 10:45 Uhr, Raum St. Petersburg
ESSBASE und die OBIEE 11g – Aufbruch zu „echten“ OLAP-Analysen 12:00 bis 12:45 Uhr, Raum Helsinki	Oracle BAM – Die unentdeckten Möglichkeiten 13:00 bis 13:45 Uhr, Raum Oslo	Versteckte Schätze in BPM&SOA Suite 11g – gesammelte Projekterfahrungen 10:00 bis 10:45 Uhr, Raum Oslo
SOA Continuous Integration 12:00 bis 12:45 Uhr, Raum Riga	Neues zur Oracle Lizenzierung 15:00 bis 15:45 Uhr, Raum Kopenhagen	Unterbrechungsfreies Reporting: Hochverfügbarkeit von OWB bis BIEE 11g 12:00 bis 12:45 Uhr, Raum Stockholm
Agile BI mit OBIEE 11g 14:00 bis 14:45 Uhr, Raum Helsinki		Oracle Resource Management 13:00 bis 13:45 Uhr, Raum St. Petersburg
Die Crux mit dem Delta – vom Fullload zum Incremental Load 16:00 bis 16:45 Uhr, Raum Kopenhagen		Forms Legacy – ein ADF Panorama 14:00 bis 14:45 Uhr, Konferenzraum EG
Automatisiertes Konfigurationsmanagement mit Puppet 16:00 bis 16:45 Uhr, Konferenzraum EG		Desaster Recovery bei Grid Infrastructure 11.2 mit zwei Rechenzentren 15:00 bis 15:45 Uhr, Raum Hongkong
		Effizientere ETL mit Table Function 16:00 bis 16:45 Uhr, Raum Stockholm



youtube.com/opitzconsulting



slideshare.net/opitzconsulting



[@OC_WIRE](https://twitter.com/OC_WIRE)



xing.com/group-51062.460375

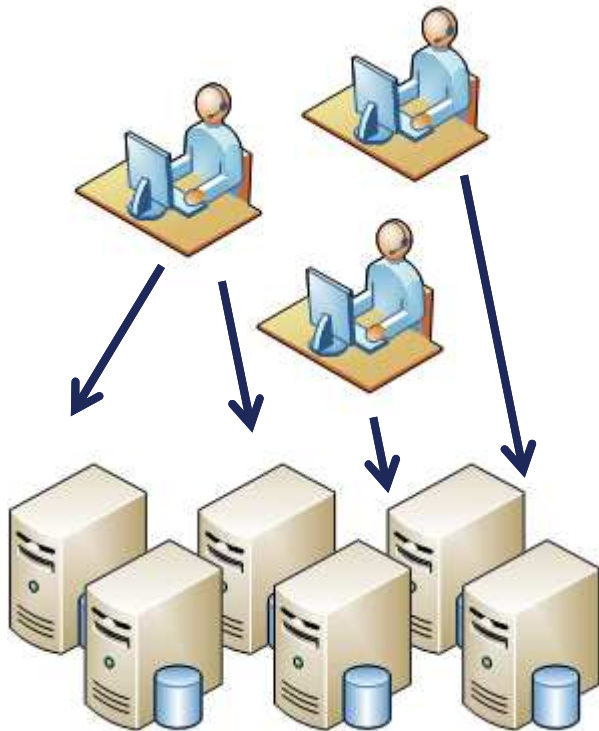
Agenda

- 1. Herausforderung an den Betrieb**
- 2. Überblick Puppet**
- 3. Puppet und Versionsverwaltung**
- 4. Fazit**

1

Herausforderungen an den Betrieb

Herausforderungen an den Betrieb



- **Komplexität**
- **Anzahl der Systeme**
- **Häufige Änderungen**
- **Verfügbarkeit**
- **Sinkende Budgets**
- **Compliance-Richtlinien**

Von der Manufaktur zur industrialisierten IT

- **Standardisierung**
- **Konsolidierung**
- **Virtualisierung**
- **Automatisierung**
- **Prozesse etablieren und optimieren**
- **Software as a Service (SaaS)**
- **Platform as a Service (PaaS)**
- **Cloud Computing**

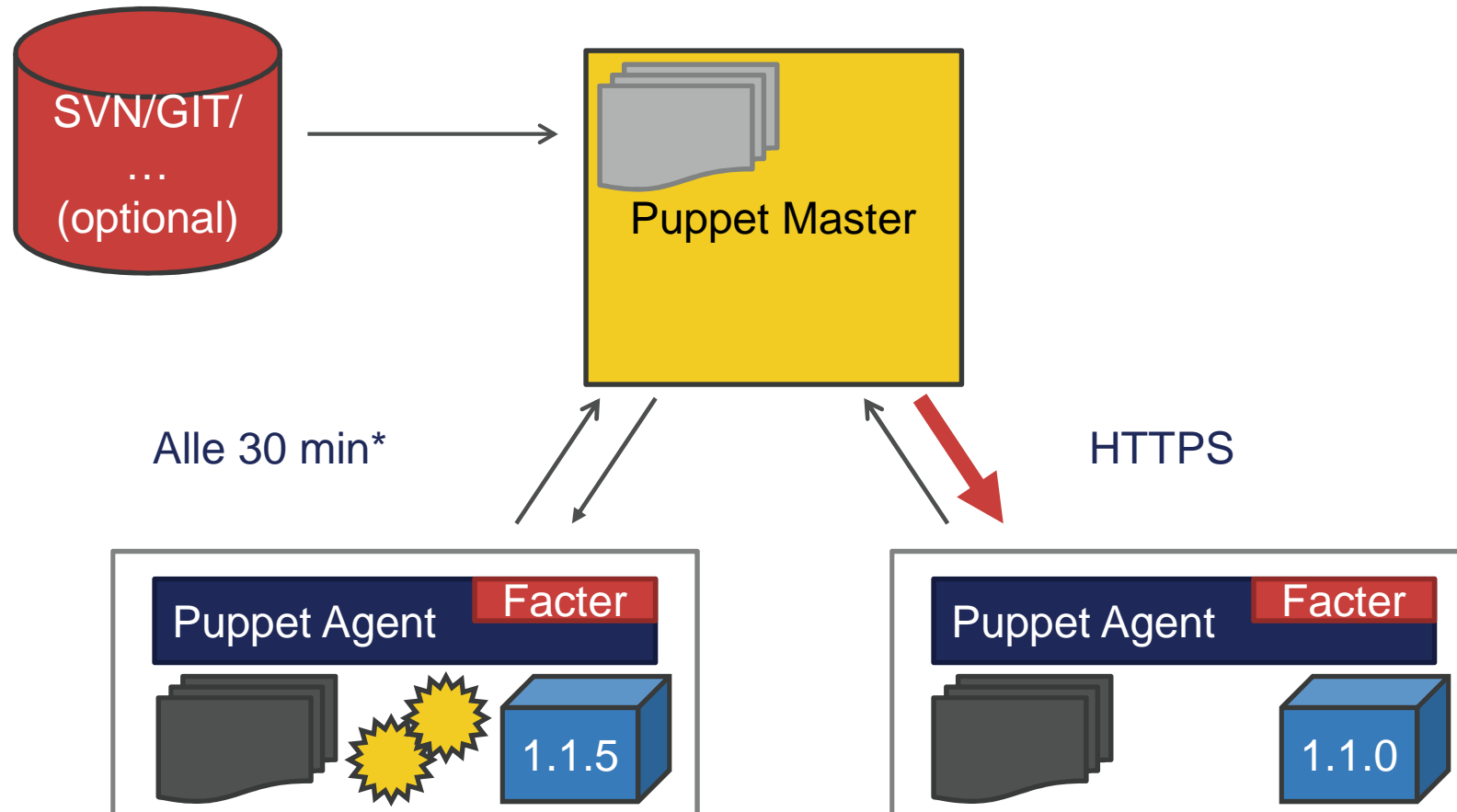
2

Überblick Puppet

Was ist Puppet?

- **Erweiterbares Framework um Server und Workstation-Konfigurationen zu administrieren**
- **Wird von PuppetLabs unter OpenSource Lizenz angeboten**
- **Erleichtert die Verwaltung von komplexen, sich verändernden Systemlandschaften**
- **Zentraler Server (Puppetmaster) verwaltet die Konfiguration**
- **Agent auf den administrierten Maschine kommuniziert mit dem Puppetmaster und sorgt für Angleichung an Zielzustand**
- **Konfigurationen werden in einer eigenen deklarativen Sprache beschrieben**

Architektur I



Unterstützte Plattformen

- **Diverse Linux Derivate**
- **Solaris, AIX, HP-UX**
- **OpenBSD, FreeBSD**
- **MacOS X**

- **(Windows) – noch experimentell**

Installationsvoraussetzungen

- **Ruby 1.8.2+ Umgebung**
- **Facter (ebenfalls von PuppetLabs)**
- **Port 8140* offen auf dem Puppetmaster**
- **Optional: Versionsverwaltung (Subversion, Git,...)**
- **Optional: Mongrel oder Apache Webserver mit Passenger Modul**
- **Kommerzielles Paket „Puppet Enterprise“ enthält alle benötigten Pakete**

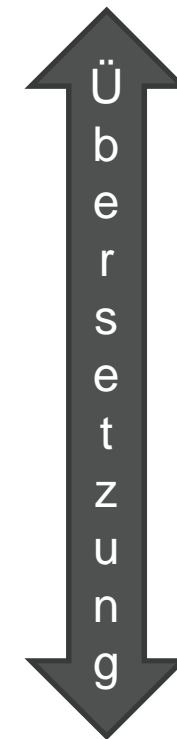
Architektur II

Deklaration

Configuration Language

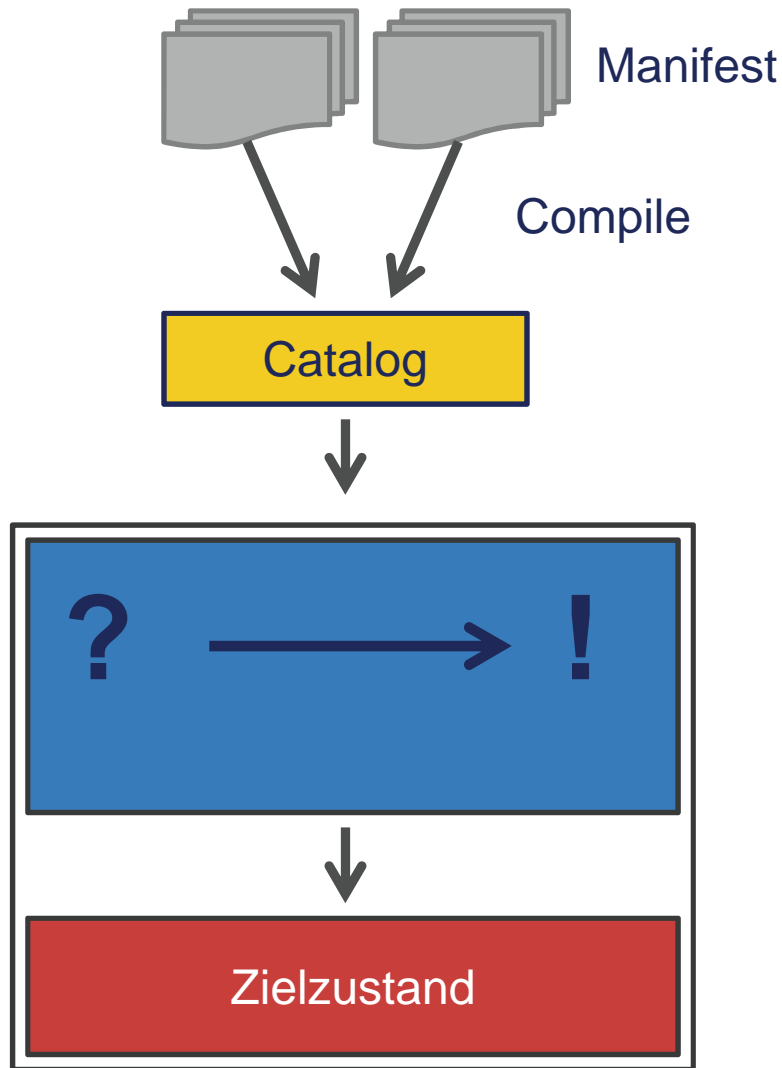
Transaction Layer

Resource Abstraction Layer



Umsetzung

Architektur III



Manifest wird zu Catalog
compiliert und an Agent
geschickt

Zustand wird ermittelt
und mit Catalog
verglichen

Abweichungen vom
definierten Zustand
werden angeglichen

Resource

- **Jede Konfigurationseinheit ist eine Resource:**
 - Useraccount
 - Datei
 - Verzeichnis
 - Softwarepaket
 - Service
 - ...

- **Jede Resource wird durch Eigenschaften definiert:**
 - Name
 - Spezifische Parameter => Werte Paare
 - Abhängigkeiten

- **Gleichartige Ressourcen werden als „Type“ bezeichnet**

Manifest

- Puppet Konfigurationsdatei
- Beschreibt die erwünschte Konfiguration
- Enthält die Ressourcendeklaration
- Manifest anwenden mit:

```
puppet apply <manifest_name.pp>
```


Beispiel: Resource für Benutzerkonto

```
user { ,schulz':  
  home => '/home/schulz',  
  uid => '500',  
  ensure => 'present',  
  gid => '501',  
  shell => '/bin/zsh',  
}
```

Das letzte Komma ist kein Tippfehler!

Beispiel: Resource für Datei

```
file { '/etc/ssh/sshd_config':  
  ensure => file,  
  owner  => root,  
  group  => root,  
  mode   => 600,  
  source => '/root/learning-manifests/sshd_config',  
}
```

Beispiel: Dienst neu starten nach Änderung

```
file { '/etc/ssh/sshd_config':  
  ...  
}  
  
service { 'sshd':  
  ensure => running,  
  enable => true,  
  hasrestart => true,  
  hasstatus => true,  
  subscribe => File['/etc/ssh/sshd_config'],  
}
```

Beispiel: Dienst neu starten nach Änderung

```
[root@learn ~]# puppet apply my_sshd_config.pp  
  
notice:  
/Stage[main]//File[/etc/ssh/sshd_config]/content: content changed '{md5}cc3dcaa54b79d442dcdd9ae3e03d3fe0' to '{md5}54180acf6ea104931a8440e3ae9d89fc'  
notice: /Stage[main]//Service[sshd]: Triggered 'refresh' from 1 events
```

Templates

- Spezifische Details in Config-Dateien werden durch Variablen ersetzt
- Variablen werden beim Schreiben der Datei auf dem Client durch die tatsächlichen Werte ersetzt
- Variablen können nach bestimmen Regex-Mustern gesetzt werden oder spezifisch für einzelne Server – oder durch Factor

Beispiel:

- Im Manifest: `$heapsize = 1000`
- In Template: `JAVA_OPTS="-Xmx<%= heapsize %>M"`

Class

- Immer wieder benötigte Konfigurationen können als Class zusammengefasst werden:

```
class myclass{  
  package {...}  
  service {...}  
  file {...}  
  user {...}  
  ...  
}
```

3

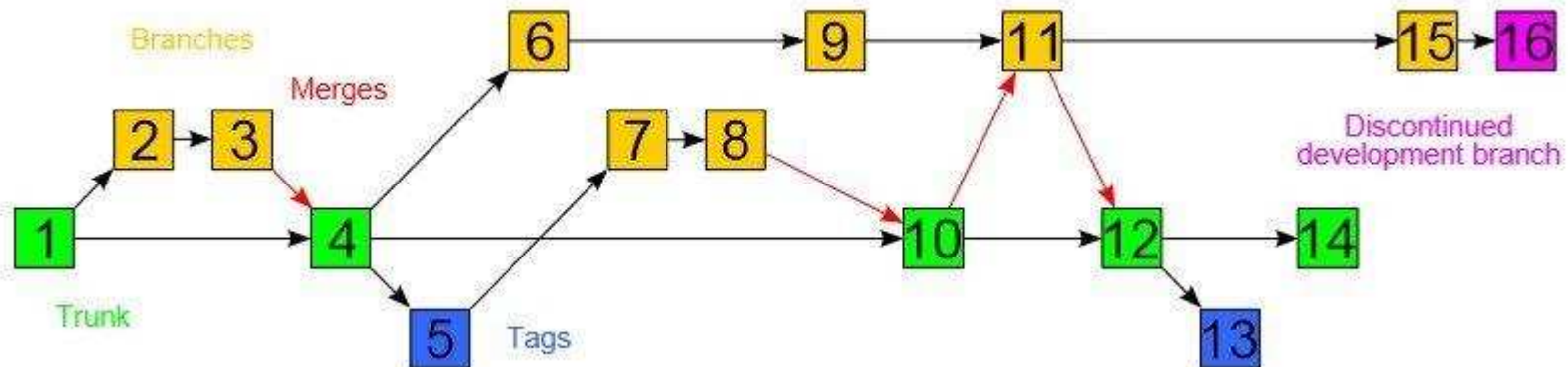
Puppet und Versionsverwaltung

Zusammenspiel mit Versionierungssystemen

- **Optional** können die Manifeste in Systemen wie Subversion oder Git verwaltet werden
- **Dokumentierte Versionshistorie**
- **Wichtige Metadaten:**
 - Wer?
 - Wann?
 - Was genau?
 - Warum?
- **Unterstützt die Trennung unterschiedlicher Konfigurationen**
- **Keine Verteilung von Änderungen ohne Commit!**

Trunk, Tags, Branches, Merging

- Trunk: sog. Mainline, Stand der Produktion
- Tags: Markierung von wichtigen Meilensteinen
- Branch: Abgeleiteter Entwicklungszweig
- Merging: Branch wird in Trunk integriert



Quelle: http://de.wikipedia.org/wiki/Apache_Subversion

4

Fazit

Noch ein Tool?

Vorteile

- **Nachvollziehbarkeit**
- **Reproduzierbarkeit**
- **Dokumentation**
- **Automatisierung**
- **Wiederverwendbarkeit**
- **Skalierbarkeit**

Nachteile

- **Einarbeitung**
- **Adminserver**
- **Ggf. SVN Server**
- **Anpassung der Betriebsprozesse**

Welche Betriebsprozesse kann Puppet unterstützen?

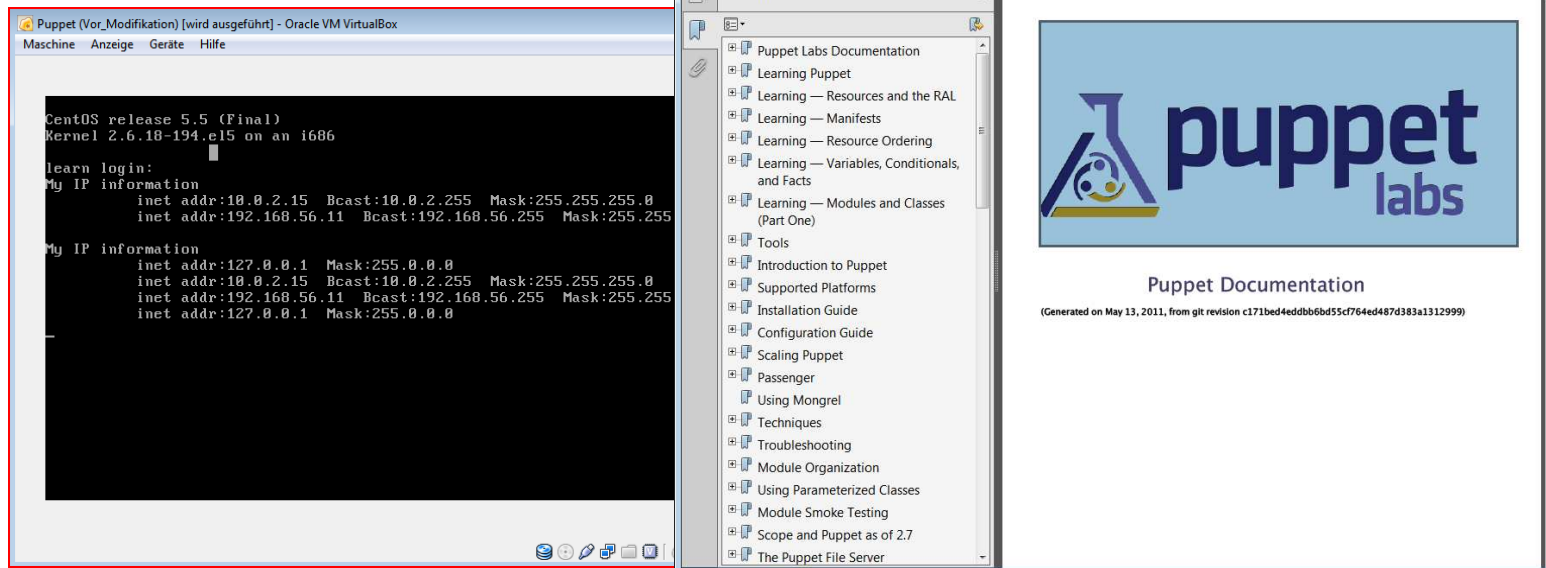
- **Provisionierung von neuen Systemen**
- **Systemhärtung**
- **Release Management**
- **Change Management**
- **Einhaltung von Comliancerichtlinien**

Puppet Good Practice

- **Standards für Serverbenennung**
- **Standards für Serversetup**
- **Verwendung von Templates**
- **Verwendung von Subversion, Git**
- **Testen in Testumgebung**

Einstieg in Puppet

- Der Hersteller PuppetLabs bietet verschiedene Materialien zum Einstieg an:
 - Puppet Dokumentation als PDF und Webseite
 - "Learning Puppet VM"
 - Wiki
 - Patternsammlung



Kontakt

Daniel Schulz

Solution Architect

OPITZ CONSULTING Gummersbach GmbH

Kirchstr. 6 | 51647 Gummersbach

Tel. +49 (2261) 60 01-0

daniel.schulz@opitz-consulting.com

