

GWDG NACHRICHTEN 06|18

Rechenzentrumsneubau

BExIS++

Research Data Alliance

Backup von
StorNext-Filesystemen

Continuous Delivery

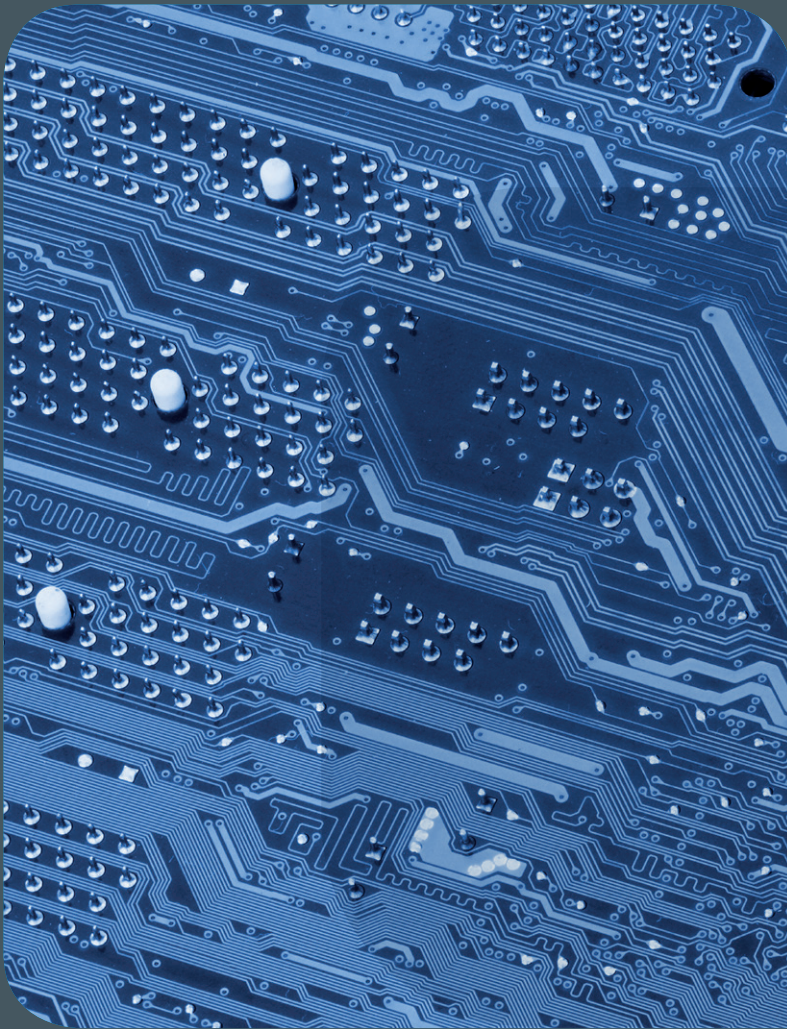
ZEITSCHRIFT FÜR DIE KUNDEN DER GWDG

A central graphic on a blue background. A large white cloud contains the text "Data Science" in bold blue letters. Surrounding the cloud are various white icons connected by lines: a laptop with a magnifying glass, a database cylinder, a cloud with arrows, a server rack, a code block with binary digits (0101, 1010, 0010), a server rack with a magnifying glass, a line graph with a magnifying glass, a magnifying glass over a bar chart, a server rack with a magnifying glass, a bar chart with an upward arrow, a cloud with a database cylinder, a pie chart, a server rack, a bar chart with an upward arrow, and a cloud with a database cylinder.

**Data
Science**

GWDG

Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen



GWGD NACHRICHTEN

06|18 Inhalt

.....

**4 Grundsteinlegung für den Neubau des
gemeinsamen Rechenzentrums 6 Prüfen und
Speichern beim Batch-Upload von BEXIS++
9 Research Data Alliance – A Global Initiative for
Standards in Research Data 10 Kurz & knapp
11 Verbesserungen beim Backup von
StorNext-Filesystemen 15 Continuous Delivery
für das Kundenportal der GWGD
24 Kurz & knapp 25 Personalia 26 Kurse**

Impressum

.....

Zeitschrift für die Kunden der GWGD

ISSN 0940-4686
41. Jahrgang
Ausgabe 6/2018

Erscheinungsweise:
monatlich

www.gwdg.de/gwdg-nr

Auflage:
550

Fotos:
© ryanking999 - Fotolia.com (1)
© fotogestoerber - Fotolia.com (8)
© Rawpixel - Fotolia.com (9)
© alain wacquier - Fotolia.com (11)
© MPLIbpc-Medienservice (3, 25)
© Gebäudemanagement der Universität
Göttingen (4)
© Universität Göttingen / Christoph
Mischke (5)
© GWGD (2, 5, 26)

Herausgeber:
Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen
Am Faßberg 11
37077 Göttingen
Tel.: 0551 201-1510
Fax: 0551 201-2150

Redaktion:
Dr. Thomas Otto
E-Mail: thomas.otto@gwdg.de

Herstellung:
Franziska Schimek
E-Mail: franziska.schimek@gwdg.de

Druck:
Kreationszeit GmbH, Rosdorf



Prof. Dr. Ramin Yahyapour
ramin.yahyapour@gwdg.de
0551 201-1545

Liebe Kunden und Freunde der GWDG,

eine Grundsteinlegung ist ein Anlass zum Feiern. Am 29. Mai stellt sie einen Meilenstein in der Planung und Umsetzung unseres neuen Rechenzentrums am Göttingen Campus dar – eine gemeinsame Maßnahme von Universität und Universitätsmedizin Göttingen und der Max-Planck-Gesellschaft mit Unterstützung des Landes Niedersachsen. Sie liefert die dringend benötigte Grundlage für eine hochmoderne und effiziente IT-Infrastruktur sowie neue Büros für die Mitarbeiterinnen und Mitarbeiter der beteiligten IT-Einrichtungen.

Bereits im Herbst 2017 begannen die Bauarbeiten an unserem neuen künftigen Standort am Burckhardtweg, ca. 1 km vom Faßberg entfernt. Mittlerweile ist die erste Etage des Bürotraktes schon fertiggestellt. In einer Welt, in der man häufig nur mit digitalen Dokumenten zu tun hat, ist es erfrischend, wenn man zur Ausnahme die Ergebnisse der Arbeit physisch greifbar sieht. Die weitere Fertigstellung des Gebäudes wird aufgrund der komplexen Technik noch ca. zwei Jahre benötigen. Da wird es mit Richtfest und Inbetriebnahme noch weitere Gelegenheiten zum Feiern geben. Wir drücken die Daumen, dass die Umsetzung möglichst problemlos verläuft.

Wir werden in den GWDG-Nachrichten hierzu berichten. Bis dahin wünsche ich Ihnen viel Freude beim Lesen dieser Ausgabe.

Ramin Yahyapour

GWDG – IT in der Wissenschaft



Grundsteinlegung für den Neubau des gemeinsamen Rechenzentrums

Text:
Presseinformation Nr. 119 der
Georg-August-Universität Göttingen
vom 29.05.2018 (mit kleinen Ände-
rungen)

Am 29. Mai 2018 fand auf dem Nordcampus der Universität Göttingen die feierliche Grundsteinlegung für den Bau des neuen gemeinsamen Rechenzentrums für den Wissenschaftsstandort Göttingen statt. Dort entsteht in einem ersten Bauabschnitt bis 2020 für rund 38,5 Mio. Euro ein Gebäudekomplex mit Maschinenraum und Büroflächen, der den Wissenschaftsstandort Göttingen in seiner internationalen Konkurrenzfähigkeit stärken soll.

Das momentan größte Neubauprojekt der Universität Göttingen nimmt Form an: Seit Anfang des Jahres entsteht auf dem Nordcampus der Bau des neuen gemeinsamen Rechenzentrums für den Wissenschaftsstandort Göttingen. Neben der Universität sind daran die Universitätsmedizin Göttingen (UMG) und die Max-Planck-Gesellschaft (MPG) beteiligt. Das Rechenzentrum der Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) stellt künftig allen beteiligten Einrichtungen zusätzliche Rechen- und Speicherkapazitäten für Forschung, Lehre, gemeinsames Arbeiten, Krankenversorgung und Administration zur Verfügung. Im ersten Bauabschnitt entsteht am Burckhardtweg 4 bis 2020 zunächst ein gemeinsam genutzter Gebäudekomplex mit Maschinenraum und Büroflächen. Die Kosten liegen bei rund 38,5 Millionen Euro.

Der Niedersächsische Minister für Wissenschaft und Kultur, Björn Thümler, sagte heute (29. Mai 2018) bei der Grundsteinlegung: „Mit seiner Digitalisierungsoffensive stärkt das Ministerium für Wissenschaft und Kultur die Hochschulen als Motoren der Digitalisierung. Dass heute der Grundstein für ein neues gemeinsames Rechenzentrum gelegt wurde, das Universität, Universitätsmedizin und Max-Planck-Gesellschaft mit zukunftsweisender IT-Infrastruktur versorgen wird, ist auf diesem Weg ein wichtiger Meilenstein.

„In idealer Lage entsteht hier ein innovativer Neubau auf der Höhe der Zeit.“

New joint Data Centre on Göttingen Campus

Göttingen University's currently biggest new construction project is starting to take shape: Since the beginning of the year, the new joint Data Centre for Göttingen's science hub has been under construction on the North Campus. In addition to the University itself, the University Medical Centre Göttingen (UMG) and the Max Planck Society (MPG) are all involved. The Data Centre of the Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG) will in future provide all stakeholder institutions with additional computing and storage capacities for research, teaching, mutual collaboration, medical care and administration. During the first building phase, it is planned that a joint complex will be built on Burckhardtweg 4 by 2020, complete with a machine room and office space. Costs have amounted to approximately € 38.5 million.



Grundsteinlegung des gemeinsamen Rechenzentrums am Göttingen Campus

„Das gemeinsame Rechenzentrum ist ein weiterer Meilenstein in der Erfolgsgeschichte des Göttingen Campus“, erklärte Universitätspräsidentin Prof. Dr. Ulrike Beisiegel. „Innovative und leistungsfähige IT- und Informationsinfrastrukturen sind mittlerweile in allen wissenschaftlichen Disziplinen unerlässlich, und die Anforderungen an sie werden weiter wachsen. Optimale Rahmenbedingungen für Forschung und Lehre bereitzustellen ist deshalb eine unserer zentralen Aufgaben.“

„Die UMG freut sich darüber, dass in absehbarer Zeit die alten IT-Räumlichkeiten und -Strukturen aus dem Zentralgebäude durch ein neues, hochmodernes und innovativ gedachtes Rechenzentrum abgelöst werden“, so Dr. Sebastian Freytag, Vorstand Wirtschaftsführung und Administration der Universitätsmedizin Göttingen. „Für die Gesamtplanung der UMG ist dies ein ganz wichtiger Baustein im Blick auf die großen Herausforderungen an die digitale Zukunftsfähigkeit der Göttinger universitären Medizin in Krankenversorgung, Forschung und Lehre.“

Prof. Dr. Martin Stratmann, Präsident der Max-Planck-Gesellschaft, betonte: „Nicht zuletzt wegen des rasanten Fortschritts im Bereich Maschinelles Lernen wird die Daten-getriebene Forschung künftig eine noch größere Rolle spielen. Hier beste Performance zu



Teilnehmer an der Grundsteinlegung (von links): Dr. Sebastian Freytag, Vorstandsmitglied der Universitätsmedizin Göttingen, Prof. Dr. Stephan Herminghaus, Direktor am Max-Planck-Institut für Dynamik und Selbstorganisation, Prof. Dr. Ulrike Beisiegel, Präsidentin der Universität Göttingen, Prof. Dr. Ramin Yahyapour, Geschäftsführer der Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen, Björn Thümler, Niedersächsischer Minister für Wissenschaft und Kultur, Rolf-Georg Köhler, Oberbürgermeister der Stadt Göttingen

bieten zählt zu den wichtigsten Schlüsselfaktoren für erfolgreiche Wissenschaft. Das macht das Rechenzentrum der Universität Göttingen zu einer ganz wesentlichen Zukunftsinvestition.“

Das Rechenzentrum ist darauf ausgelegt, den Wissenschaftsstandort Göttingen in seiner internationalen Konkurrenzfähigkeit zu stärken. Künftig laufen dort die Forschungsinfrastrukturen für das Hochleistungsrechnen, das Daten- und Bildmanagement sowie die Langzeitarchivierung bei wachsenden Anforderungen an die Verfügbarkeit der Daten zusammen. Aus dem Projekt ergeben sich unterschiedliche Synergien: So soll beispielsweise die im Rechenzentrum durch die Rechnerleistung anfallende Abwärme genutzt werden, um die südlich des Burckhardtwegs geplanten Gewächshäuser der Fakultät für Agrarwissenschaften zu heizen. In einem zweiten Bauabschnitt sollen die Kapazitäten des Rechenzentrums erweitert und der Bereich eResearch gestärkt werden. ■



Blick über die Baustelle des gemeinsamen Rechenzentrums auf dem Nordcampus der Universität Göttingen (Aufnahme am Tag der Grundsteinlegung)

Prüfen und Speichern beim Batch-Upload von BExIS++

Text und Kontakt:

Thorsten Hindermann
thorsten.hindermann@gwdg.de
0551 201-1837

Nachdem sich schon zwei Artikel in den GWDG-Nachrichten 8-9/2016 und 8-9/2017 mit speziellen Aspekten des Batch-Uploads für das DFG-Projekt „BExIS++“, an dem die GWDG beteiligt ist, befasst haben, werden in diesem dritten abschließenden Artikel noch zwei wesentliche interne Routinen vorgestellt: das Prüfen von ausgewählten Excel-Dateien, die zu einem Dataset gehören, und das Speichern der überprüften Dateien.

PRÜFEN

Eine der zwei zentralen Funktionen im Batch-Upload ist das Prüfen von Excel-Dateien gegen ein ausgewähltes BExIS++-Dataset. Erst wenn die Prüfung das Boolesche Ergebnis „Wahr“ ergibt, wird die Datei bzw. werden die Dateien in der Auswahlliste im Batch-Upload-Formular angezeigt.

Eine entscheidende Quellzeile ist in Zeile 1091 der Quellzeilen-Auflistung „Prüfen“ zu sehen:

```
if (_selectMultipleFilesViewModel.SelectedDatasetId > 0)
```

Wenn die abgefragte Eigenschaft *SelectedDatasetId* größer 0 ist, ist im Batch-Upload-Formular zuvor ein existierendes Dataset ausgewählt worden. Sodann beginnt die Überprüfung der Excel-Datei, repräsentiert durch die Struktur *bfi* vom Typ *BasicFileInfo*. Die gesamte Prüfung ist in den Zeilen 1093 bis 1152 nachzuvollziehen. Ist der eingangs abgefragte Wert der Eigenschaft gleich 0, dann wird das Boolesche Ergebnis „Falsch“ zurückgegeben. In dem Formular werden dann alle Excel-Dateien, egal welchem Dataset diese zugeordnet sind, angezeigt, deren Überprüfung den Rückgabewert „Wahr“ ergeben hat.

In dieser Phase der Implementierung/Anforderung war die Auswertung von ASCII-Dateien nicht vorgesehen, sodass die Zeilen 1126 bis 1148 derzeit nur minimale Verarbeitungsaufrufe zeigen, die sich derzeit nur auf ein Öffnen und Schließen von etwaigen ASCII-Dateien beschränken.

SPEICHERN

Ähnlich verhält sich die Sachlage beim Speichern der Excel-Dateien. ASCII-Dateien werden auch hier nicht berücksichtigt. In dieser Funktion wird ebenfalls abgefragt, ob zuvor ein Dataset ausgewählt wurde:

```
if (_selectMultipleFilesViewModel.SelectedDatasetId > 0)
```

Wenn nicht, wird das Boolesche Ergebnis „Falsch“ zurückgeliefert. Ist dagegen ein Dataset ausgewählt worden, beginnt die Verarbeitungsroutine ab der Zeile 1174 bis Zeile 1217. Zu beachten sind die Zeilen 1206 bis 1210 (siehe die Quellzeilen am Ende dieser Seite).

Diese Schlaf-Funktion war notwendig geworden, weil es in Testläufen bei der Verarbeitung immer wieder vorkam, dass die Dateien nicht ordnungsgemäß verarbeitet werden konnten. Es braucht seine Zeit, bis die aus den Excel-Dateien gelesenen Datensätze in die entsprechenden Tabellen des Datenbank-Servers ordnungsgemäß geschrieben worden sind. Deshalb ist die Abfrage in Zeile 1207 wichtig, ob mehr als eine Datei verarbeitet wird. Wenn ja und die Schlafzeit der Variable *sleep* größer 0 ist, wird die entsprechende Zeitspanne gewartet. In dieser Zeit hat dann der Datenbank-Server die Daten in die entsprechenden Tabellen geschrieben.

Die Zeitspanne in der Variablen *Sleep* kann von außen mit Hilfe der Konfigurations-Datei *web.config* von einem Server-Verwalter eingestellt werden, ohne dass die Webanwendung nach jeder neuen Einstellung neu erstellt werden muss. Die dazu notwendigen Zeilen sind 1166 bis 1169:

```
// Get the value from the web.config
var valueTts = UploadWizardHelper.GetWebConfKeyValue(„TTS“);
// Change string into int32 value
var sleep = Int32.Parse(valueTts);
```

In der BExIS++-Version, mit dem der Batch-Upload entwickelt wurde, hatte die Funktion *UndoCheckOutDataset* noch so ihre Tücken. Schlug die Verarbeitung einer Excel-Datei fehl, konnte in der Version diese Funktion den Ursprungszustand, der vor

```
// If there a (many) files to store into the BExIS++ database for the moment there is a artificial delay necessary
if (_selectMultipleFilesViewModel.ServerMultipleBasicFileInfoList.Count > 0)
    if (sleep > 0) // Check...
        Thread.Sleep(sleep); //...and sleep well for a few seconds ;-)
```

Beginn der Verarbeitung vorgelegen hat, leider nicht wiederherstellen. In der neueren Version von BEXIS++ soll dieses Manko laut den Kern-Entwicklern in Jena aber behoben worden sein.

ABSCHLUSS

Mit diesem abschließenden Artikel endet auch die Arbeit am Arbeitspaket „Batch-Upload“ für das BEXIS++-Projekt. Das

gesamte Projekt wurde die ganze Zeit auf dem Quellzeilenverwaltungs-Server der BEXIS++-Projektgruppe in Jena gespeichert. Die Information über den Abschluss ist an die Projektgruppe gegangen. Diese hat zurückgemeldet, dass sie nun die Quellzeilen des Batch-Uploads in den Haupt-Entwicklungszweig übernehmen und einbauen will. Somit ist zu erwarten, dass in einer neueren, zukünftigen BEXIS++-Version diese Funktion zur Verfügung stehen wird. ■

```

1084     /// <summary>
1085     /// Check the given dataset.
1086     /// </summary>
1087     /// <param name="bfi">BasicFileInfo structure </param>
1088     /// <returns>True if the dataset matches the selected</returns>
1089     private bool CheckDataSet(BasicFileInfo bfi)
1090     {
1091         if (_selectMultipleFilesViewModel.SelectedDataSetId > 0)
1092         {
1093             ExcelReader excelReader = new ExcelReader();
1094             AsciiReader asciiReader = new AsciiReader();
1095             AsciiFileReaderInfo asciiFileReaderInfo = new
1096                 AsciiFileReaderInfo();
1097             FileStream fileStream = null;
1098             var aFile = bfi.Uri + "\\\" + bfi.Name;
1099             string[] fextSplit = new[] {string.Empty};
1100             fextSplit = UploadWizardHelper.GetWebConfKeyValue
1101                 ("StructFExt").Split(',');
1102             var fext = "." + bfi.Name.Split('.').Last();
1103             DatasetManager dm = new DatasetManager();
1104             long id = _selectMultipleFilesViewModel.SelectedDataSetId; //
1105             DataSet_ID
1106             long dsid = dm.GetDataset(id).DataStructure.Id; //
1107             DataStructure_ID
1108             DataStructureManager dsm = new DataStructureManager();
1109             StructuredDataStructure sds =
1110                 dsm.StructuredDataStructureRepo.Get(dsid);
1111             dsm.StructuredDataStructureRepo.LoadIfNot(sds.Variables);
1112             List<Error> tempErrorList = new List<Error>();
1113
1114             //try .. catch - construct
1115             try
1116             {
1117                 fileStream = excelReader.Open(aFile);
1118                 if (IsSupportedFileExtension(fext, fextSplit) == true)
1119                 {
1120                     excelReader.ValidateFile(fileStream, aFile, sds, id);
1121                     // _selectMultipleFilesViewModel.ErrorList =
1122                     excelReader.ErrorMessages;
1123                     tempErrorList = excelReader.ErrorMessages;
1124                     fileStream.Close();
1125                 }
1126                 else
1127                 {
1128                     fileStream.Close();
1129                     return false; // no supported files extension n the
1130                     first test, no further processing necessary
1131                 }
1132             }
1133             catch (Exception e)
1134             {
1135                 try
1136                 {
1137                     fileStream.Close(); // close filestream from first try
1138                     fileStream = asciiReader.Open(aFile);
1139                     if (IsSupportedFileExtension(fext, fextSplit) == true)
1140                     {
1141                         //asciiReader.ValidateFile(fileStream, aFile,
1142                         asciiFileReaderInfo, sds, id);
1143                         // _selectMultipleFilesViewModel.ErrorList =
1144                         excelReader.ErrorMessages;
1145                         fileStream.Close();
1146                     }
1147                     else
1148                     {
1149                         fileStream.Close();
1150                     }
1151                 }
1152                 catch (Exception exception)
1153                 {
1154                     fileStream.Close(); // close filestream from previous
1155                     try
1156                     {
1157                         return false;
1158                     }
1159                 }
1160             }
1161             if (tempErrorList.Count == 0) // don't use the global ErrorList
1162                 return true;
1163             else
1164                 return false;
1165         }
1166         else
1167         {
1168             return false; // No dataset selected the complete check fails
1169         }
1170     }

```

Quellzeilen-Auflistung „Prüfen“

```

1159     /// <summary>
1160     /// Save the given dataset
1161     /// </summary>
1162     /// <param name="bfi"></param>
1163     /// <returns>True if the dataset could be stored in the database.</
1164     returns>
1165     private bool SaveDataSet(BasicFileInfo bfi)
1166     {
1167         // Get the value from the web.config
1168         var valueTts = UploadWizardHelper.GetWebConfKeyValue("TTS");
1169         // Change string into int32 value
1170         var sleep = Int32.Parse(valueTts);
1171
1172         if (_selectMultipleFilesViewModel.SelectedDataSetId > 0)
1173         {
1174             ExcelReader excelReader = new ExcelReader();
1175             AsciiReader asciiReader = new AsciiReader();
1176             AsciiFileReaderInfo asciiFileReaderInfo = new
1177                 AsciiFileReaderInfo();
1178             FileStream fileStream = null;
1179             var aFile = bfi.Uri + "\\\" + bfi.Name;
1180             string[] fextSplit = new[] {string.Empty};
1181             fextSplit = UploadWizardHelper.GetWebConfKeyValue
1182                 ("StructFExt").Split(',');
1183             var fext = "." + bfi.Name.Split('.').Last();
1184             DatasetManager dm = new DatasetManager();
1185             long id = _selectMultipleFilesViewModel.SelectedDataSetId; //
1186             DataSet_ID
1187             long dsid = dm.GetDataset(id).DataStructure.Id; //
1188             DataStructure_ID
1189             DataStructureManager dsm = new DataStructureManager();
1190             StructuredDataStructure sds =
1191                 dsm.StructuredDataStructureRepo.Get(dsid);
1192             dsm.StructuredDataStructureRepo.LoadIfNot(sds.Variables);
1193
1194             //try .. catch - construct
1195             try
1196             {
1197                 DataRow[] rows = new DataRow[] { };
1198                 Dataset ds = dm.GetDataset(id);
1199                 dm.CheckOutDatasetIfNot(id, GetUserNameOrDefault());
1200                 var workingCopy = dm.GetDatasetWorkingCopy(id);
1201
1202                 fileStream = excelReader.Open(aFile);
1203                 if (IsSupportedFileExtension(fext, fextSplit) == true)
1204                 {
1205                     rows = excelReader.ReadFile(fileStream, aFile, sds,
1206                     id).ToArray();
1207                     _selectMultipleFilesViewModel.ErrorList =
1208                     excelReader.ErrorMessages;
1209                     dm.EditDatasetVersion(workingCopy, rows, null, null);
1210                     //dm.CreateContentDescriptor("CSG", "", "file:///C:/
1211                     Users/hindermath/Documents/BEXIS", 1, workingCopy);
1212                     dm.CheckInDataset(id, "Upload data from Batch Upload
1213                     wizard", GetUserNameOrDefault());
1214                     //dm.UndoCheckoutDataset(id, GetUserNameOrDefault
1215                     ()); //Hat ein Problem. Mit Vorsicht genießen und nur im
1216                     Catch-Zweig einer Try/Catch nutzen!!!!
1217                     // If there a (many) files to store into the BEXIS++
1218                     database for the moment there is a artificial delay
1219                     necessary
1220                     if
1221                     (
1222                     _selectMultipleFilesViewModel.ServerMultipleBasicFileInfoLi
1223                     st.Count > 0)
1224                     if (sleep > 0) // Check...
1225                         Thread.Sleep(sleep); //...and sleep well for a
1226                         few seconds ;-)
1227                     fileStream.Close(); // close filestream after successful
1228                     CheckInDataset
1229                 }
1230                 else
1231                 {
1232                     fileStream.Close(); // close filestream from first try
1233                     return false;
1234                 }
1235             }
1236             catch (Exception e)
1237             {
1238                 try
1239                 {
1240                     dm.UndoCheckoutDataset(id, GetUserNameOrDefault()); //
1241                     Vorherigen, unbearbeiteten, Checkout gleich wieder
1242                     rückgängig machen
1243                     fileStream.Close(); // close filestream from first try
1244                     _selectMultipleFilesViewModel.ErrorList.Add(new Error
1245                     (ErrorType.Dataset, "The current Excel file has not been
1246                     processed!"));

```

Quellzeilen-Auflistung „Speichern“ (Teil 1)

```

1225         //fileStream = asciiReader.Open(aFile);
1226         // ASCII file processing goes here in the future
1227         //fileStream.Close();
1228     }
1229     catch (Exception exception)
1230     {
1231         fileStream.Close(); // close filestream from previous
1232     try
1233         return false;
1234     }
1235     if (_selectMultipleFilesViewModel.ErrorList.Count == 0)
1236         return true;
1237     else
1238         return false;
1239 }
1240 else
1241 {
1242     return false; // No dataset selected - the complete save fails
1243 }
1244 }
1245 }

```

Quellzeilen-Auflistung „Speichern“ (Teil 2)

Checking and Saving in the Batch Upload of BEXIS++

After two articles in the GWDG News 8-9/2016 and 8-9/2017 have already dealt with special aspects of the batch upload for the DFG project „BEXIS++“, in which the GWDG is involved, two essential internal routines are presented in this third final article: checking selected Excel files belonging to a dataset and saving the checked files.



Mailinglisten

Mailversand leicht gemacht!

Ihre Anforderung

Sie möchten per E-Mail zu oder mit einer Gruppe ausgewählter Empfänger kommunizieren, auch außerhalb Ihres Instituts. Sie möchten selbstständig eine Mailingliste verwalten, z. B. Empfänger hinzufügen oder entfernen. Bei Bedarf sollen sich auch einzelne Personen in diese Mailingliste einschreiben dürfen.

Unser Angebot

Wir bieten Ihnen einen Listserver, der zuverlässig dafür sorgt, dass Ihre E-Mails an alle in die Mailingliste eingetragenen Mitglieder versendet werden. Die E-Mails werden automatisch archiviert. Das Archiv kann von allen Mitgliedern der Liste nach Schlagwörtern durchsucht werden. Die Anzahl Ihrer Mailinglisten ist unbegrenzt.

Ihre Vorteile

- > Leistungsfähiges ausfallsicheres System zum Versenden von vielen E-Mails
- > Sie senden Ihre E-Mail lediglich an eine Mailinglisten-Adresse, die Verteilung an die Mitglieder der Mailingliste übernimmt der Listserver.

- > Listenmitglieder können an diese E-Mail-Adresse antworten. Eine Moderationsfunktionalität ist verfügbar, mit der Sie die Verteilung einer E-Mail genehmigen können.
- > Voller administrativer Zugriff auf die Einstellungen der Mailingliste und der Listenmitglieder
- > Obsolete E-Mail-Adressen werden vom System erkannt und automatisch entfernt.
- > Wenn Ihre E-Mail-Domäne bei uns gehostet wird, können Sie auch die Adresse der Mailingliste über diese Domäne einrichten lassen.

Interessiert?

Für die Einrichtung einer Mailingliste gibt es zwei Möglichkeiten: Zum einen als registrierter Benutzer der GWDG im Selfservice über das Kundenportal der GWDG und zum anderen, indem Sie bitte eine entsprechende E-Mail an support@gwdg.de senden, die die Wunsch-E-Mail-Adresse der Liste sowie die E-Mail-Adresse der Person, die die Liste bei Ihnen administrieren soll, enthalten sollte. Die administrativen Aufgaben sind leicht zu erlernen.

>> www.gwdg.de/maillinglisten



Research Data Alliance – A Global Initiative for Standards in Research Data

Text and Contact:
Dr. Sven Bingert
sven.bingert@gwdg.de
0551 201-2164

The 11th RDA Plenary took place from March 21st to 23rd 2018 in Berlin. Scientists, infrastructure and service providers from around the world came together for interesting presentations and discussions on research data. Also members of the Göttingen Campus joined the meeting, why we would like to give a short overview about RDA and our contributions within this community.

ABOUT RDA

The Research Data Alliance (RDA) [1] aims at building the social and technical bridges to enable open sharing of data. Data should be openly shared across technologies, disciplines, and countries. The RDA was launched in 2013 by the European Commission, the United States National Science Foundation and National Institute of Standards and Technology, and the Australian Government's Department of Innovation. Their goal was to unite the highly fragmented research data landscape and to develop solutions for open sharing of data. Every researcher, institute head or anybody working with research data can join and participate in RDA, but needs to agree to the guiding principles of openness, consensus, balance, and harmonisation. The RDA actions are community driven with a non-profit approach. As of February 2018, RDA had over 6.600 members from 135 countries collaborating together in Working and Interest Groups with the aim to create recommendations and other outputs to foster data-sharing and data-driven research.

RECOMMENDATIONS & OUTPUTS

The Working and Interest Groups are short or medium time living collaborations to find solutions that enable e.g. data sharing, exchange, and interoperability. These solutions are derived from community use cases or problems and may affect infrastructure

Research Data Alliance

Im März 2018 fand das 11. RDA Plenary, mit großer Beteiligung auch aus Göttingen, in Berlin statt. Die Research Data Alliance (RDA) besteht aus internationalen Wissenschaftlern, Infrastruktur- und Serveranbietern und wurde 2013 von der Europäischen Kommission, der United States National Science Foundation und dem National Institute of Standards and Technology sowie dem Innovationsministerium der australischen Regierung ins Leben gerufen. Mit einem agilen Konzept von kurzlebigen Arbeitsgruppen werden technische und gesellschaftliche Probleme beim internationalen Austausch von und der gemeinsamen Arbeit mit Forschungsdaten untersucht und Lösungen entwickelt. Die GWDG beteiligt sich aktiv in verschiedenen Arbeitsgruppen und adaptiert oder implementiert wichtige Ergebnisse.

Seit Anfang 2018 ist Dr. Jan Brase, Leiter der Abteilung „Forschung und Entwicklung“ der Niedersächsischen Staats- und Universitätsbibliothek Göttingen (SUB), im Vorstand des im November 2017 gegründeten Vereins RDA-DE [5], der zum Ziel hat, die Zusammenarbeit der Daten-Experten in Deutschland zu stärken und mit den Arbeiten der übergeordneten Research Data Alliance zu verknüpfen.

environments run by projects or organizations. The spread of the RDA recommendations within the research community is a key success factor.

RDA outputs can be classified in recommendations, supporting outputs and other outputs. Most important are the RDA Recommendations, as they are the official, endorsed results of a Working Group. They have undergone formal phases of discussion, comment, and decision-making, and can be compared to "specifications" or "standards". RDA Recommendations can be documents including specifications, taxonomies or ontologies, workflows, schemas and many more. To be endorsed as official RDA output, they have to meet specific criteria, e.g. open access and publicly usable. Some of the RDA Recommendations are even published as ICT Technical Specifications [2].

STRUCTURE OF RDA

The idea of RDA is set up agile and short living organizations with a clear goal. These organizations are described as Working and Interest Groups [3]:

- Working Groups are short-term (18 months) and come together to develop and implement data infrastructures, which could be tools, policy, practices and products that are adopted and used by projects, organizations, and communities. Embedded within these groups are individuals who will use the infrastructure and help in making it broadly available to the public. As of February 2018, RDA boasts 33 working groups.

- Interest groups are open-ended in terms of longevity. They focus on solving a specific data sharing problem and identifying what kind of infrastructure needs to be built. Interest Groups can identify specific pieces of work and start up a Working Group to tackle those. As of February 2018, RDA boasts 58 interest groups.

GWDC CONTRIBUTIONS

Members of the GWDC actively contribute to different RDA Working Groups (WG). Mostly in those related to Persistent Identifiers (PID), their usage and service around them, but we also take part in the WG related to Federated Identity Management and Data Management. GWDC follows the announcements of RDA outputs, especially the recommendations, and investigates to adopt or implement those in our workflows or services. E.g. the output of the "Data Type Registries (DTR)" is now implemented as a "PID Information Type Registry" [4] in order to create, manage, and use of additional metadata used for information about the data in a PID record.

REFERENCES

- [1] <https://www.rd-alliance.org>
- [2] https://ec.europa.eu/growth/industry/policy/ict-standardisation/ict-technical-specifications_en
- [3] <https://www.rd-alliance.org/groups>
- [4] <http://dtr-pit.pidconsortium.eu/>
- [5] <http://rda-deutschland.de>

Kurz & knapp

Data Science Summer School 2018

As part of the Short Term Programs of Göttingen University the 2nd Summer School on Data Science will take place from August 2nd – 16th 2018 in Göttingen. This year 36 students, mainly coming from multiple international cooperation partner institutions and from Göttingen University itself, will participate. Organized and prepared is the Summer School in cooperation of the Institute of Computer Science with Göttingen International, the Institute for Medical Informatics of UMG and the Göttingen eResearch Alliance. The Summer School will residence in the heart of the city in the "Alte Mensa" on Wilhelmsplatz.

Within the two week course the participants will be learning and (net)working intensively on data science topics assorted around the research data life cycle, e. g. on classification, data mining, language processing, data analytics, linear models, statistical methods, data visualization and ethical aspects of data science. Most of the speakers are from Göttingen University and the German-Japanese University Network HeKKSaGOn. The scientific program is rounded off by events like a welcome reception, guided tours through Göttingen and excursions.

More information under <https://www.uni-goettingen.de/en/575381.html>.

Steilen



Verbesserungen beim Backup von StorNext-Filesystemen

Text und Kontakt:

Björn Nachtwey
bjoern.nachtwey@gwdg.de
0551 201-2181

Die GWDG setzt, ebenso wie die beiden Göttinger Max-Planck-Institute für biophysikalische Chemie (MPI-BPC) und Experimentelle Medizin (MPI EM), auf Quantum StorNext als zentrales Filesystem für die Forschungsdaten. Neben zahlreichen Vorteilen offenbarte sich aber beim Backup dieser Daten ein Problem im Zusammenspiel mit Windows-Servern: Der TSM/ISP-Backup-Client identifiziert geänderte Dateien auch am Datum der letzten Änderung und greift hierfür auf Windows-Funktionen zurück. Leider hat die genutzte Funktion für Winter- und Sommerzeit unterschiedliche Werte zurückgegeben, so dass für diese StorNext-Filesysteme jeweils zur Zeitumstellung eine Vollsicherung ausgeführt wurde. Dieser Artikel beschreibt eine Lösung des Problems, die zusammen mit den Kollegen der beiden genannten Max-Planck-Institute erarbeitet und erfolgreich umgesetzt wurde.

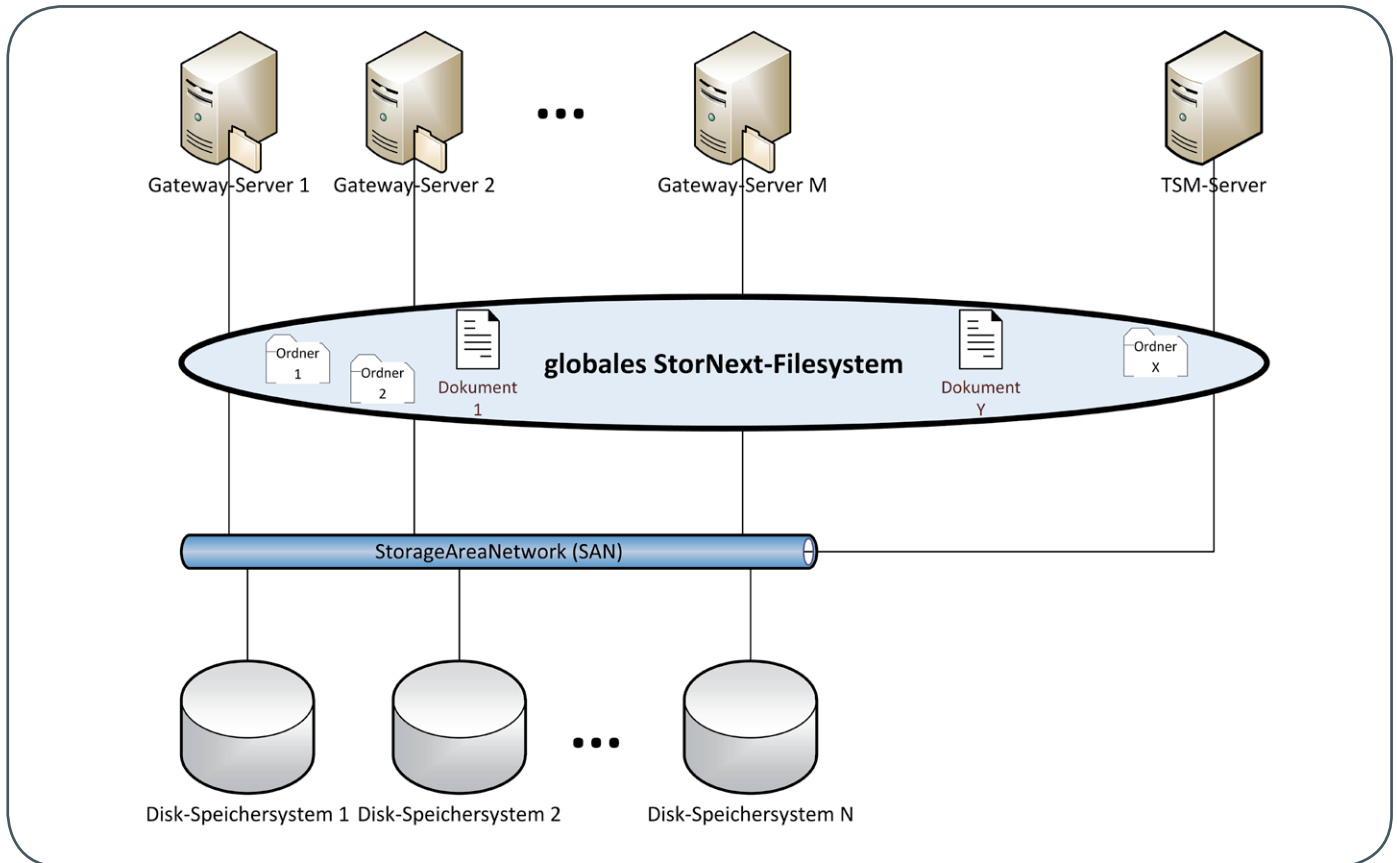
AUSGANGSLAGE UND ANSATZ ZUM SICHERN DER GROSSEN FILESYSTEME

Die Windows-StorNext-Filesysteme (SNFS) der GWDG wurden zwar zu großen Teilen zwischenzeitlich auf einen neuen NetApp-Filer migriert, bei den Fakultäten für Agrarwissenschaften, für Forstwissenschaften und Waldökologie und für Biologie und Psychologie besteht aber weiterhin der Bedarf an einer Datenablage für umfangreiche Forschungsdaten. Hierfür wurden die vormals bestehenden StorNext-Filesysteme beibehalten, die über aktuell etwa 500 Terabyte (TB) Kapazität verfügen. Am MPI für biophysikalische Chemie summiert sich die Kapazität auf rund 920 TB für die Arbeitsgruppensdaten, am MPI für Experimentelle Medizin beläuft sich der Bedarf für die Abteilungen und Arbeitsgruppen auf rund 1,2 Petabyte (PB). Allen drei Szenarien ist gemeinsam, dass der Umfang der gesicherten Datenmenge ausreichend groß genug ist, um jeweils dedizierte Backup-Server sinnvoll erscheinen zu lassen.

Ein „klassisches File-Backup“ über Proxy-Server oder sogar die Gateway-Server, die das Filesystem per CIFS bereitstellen, bedeutet für diese eine hohe zusätzliche Last. Daher wurde der

Improvements on Backing up StorNext Filesystems

GWDG as well as two local Max Planck Institutes use Quantum StorNext as a central filesystem for scientific data. Even if SNFS has many advantages one big issue occurs if the servers run on Windows: due to adjustment of the timezone for daylight saving twice a year, all data time stamps change and the backup with TSM/ISP initiates a full backup. In this article an approach to solve this problem is described taking the MPI for Biophysical Chemistry and the MPI for Experimental Medicine as an example.



1_TSM-Server als Mitglied im StorNext-Filesystem (vereinfachte Darstellung)

Ansatz gewählt, die TSM/ISP-Server selbst zum Teil des globalen Filesystems zu machen und die Backup-Clients direkt auf den TSM/ISP-Servern zu installieren (siehe Abb. 1). Mit diesem Ansatz kann die Datenübertragung zwischen Client und Server direkt über eine sogenannte „Named Pipe“ oder über eine „Shared Memory“-Schnittstelle erfolgen, also ohne Einbindung des LANs oder der entsprechenden Betriebssystemteile („IP-Stack“). Die Latenzen sind nahezu nicht vorhanden, vor allem aber wird der Durchsatz nicht von Netzwerkbandbreiten beschränkt.

Dieser Ansatz mit dem TSM/ISP-Server im StorNext-Filesystem wird für die UNIX-Filesysteme in gleichem Maße umgesetzt und die allgemeinen Aussagen sind vollständig übertragbar. Da dort aber das nachfolgend dargestellte „Zeit-Problem“ nicht auftritt, wird in diesem Artikel nur auf die Windows-Server fokussiert.

BESCHREIBUNG DES ZEIT-PROBLEMS

Für die Identifikation von Änderungen an Dateien, die zu einem Backup beim „Incremental“-Ansatz führen, nutzt TSM/ISP verschiedene Kriterien, die sich teilweise zur Beschleunigung des Backups auch abschalten lassen. Ein Kriterium, das jedoch immer ausgewertet wird, ist der Zeitpunkt der letzten Änderung. Stimmt dieser nicht mit dem Eintrag in der TSM-internen Datenbank überein, wird die betreffende Datei als geändert bewertet und erneut gesichert. Das Problem, dass sich durch die Zeitumstellung zweimal im Jahr die Uhrzeiten ändern, tritt in der Regel nicht auf, da der Zugriffszeitpunkt sowohl das Datum als auch die Uhrzeit umfassen. Ein Änderungsdatum vor der Zeitumstellung bleibt daher erhalten und stimmt mit dem Sicherungsdatum weiterhin überein.

Die unterschiedlichen Ansätze, mit denen UNIX-Systeme und Windows die Zeitzone interpretieren, sind TSM/ISP-serverseitig normalerweise nicht von Belang, da die Sicherungszeit vom Client übertragen wird und vollkommen unabhängig von der Serverzeit ist. Damit ist auch die Sicherung sowohl von Linux- als auch Windows-Clients auf einem TSM/ISP-Server des jeweils anderen Betriebssystems problemlos möglich. Bei globalen Filesystemen, die nativ sowohl UNIX/Linux- als auch Windows-Server unterstützen, muss natürlich eine Lösung für das Problem der unterschiedlichen Zeitzeonen-Interpretation gefunden werden.

Bei StorNext wird die Zeit als Metainformation sowieso in den Metadaten-Servern gespeichert; diese nutzen die sogenannte FILETIME, also die UNIX-Zeit (Sekunden seit 01.01.1970). Auf den jeweiligen Gateway-Servern interpretiert der StorNext-Client dann die FILETIME. Für Windows gibt es hierzu die Funktion „FileTimeToDosDateTime“ (siehe [https://msdn.microsoft.com/en-us/library/windows/desktop/ms724274\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms724274(v=vs.85).aspx)). Und hier liegt die Ursache für die Vollsicherungen: Mit der Europäischen Winterzeit (MEZ, CET) und der Europäischen Sommerzeit (MESZ, CEST) gibt es zwei Zeitzeonen. Die Funktion „FileTimeToDosDateTime“ rechnet die Zeit passend zur aktuellen Zeitzone aus, selbst wenn für das Datum der letzten Änderung eine andere Zeitzone galt. Nach der Zeitumstellung bekommen also alle Dateien mit einem letzten Zugriff vor der Zeitumstellung eine um eine Stunde verschobene Zugriffszeit. Vergleicht nun der TSM/ISP-Client diese mit der vor der Zeitumstellung gespeicherten Sicherungszeit, passen beide nicht zueinander und die Dateien werden erneut gesichert. Da dies bei der Zeitumstellung für ausnahmslos alle Dateien gilt, findet eine Vollsicherung statt.

IDEE DES LÖSUNGSANSATZES

Da die Ursache also im Wechsel der Zeitzonen bzw. der sich hieraus ergebenden Fehlinterpretation der FILETIME liegt, ist die Abhilfe wohl darin zu finden, dass der Backup-Client dauerhaft in einer Zeitzone verbleibt. Dies kann allerdings weder die Europäische Winter- noch Sommerzeit sein, da dann für diese Windows-Server über eine entsprechende Richtlinie die Zeitanpassung unterbunden werden müsste. Diese Richtlinie darf natürlich nur für die Backup-Server gelten und nicht Domänen-weit. Allerdings widerspricht eine solche Richtlinie der Windows-Standard-Richtlinie, dass eine Anpassung an „DayLight Saving“ zu erfolgen hat. Es wäre also eine Ausnahme zu definieren und sicherzustellen, dass diese weder aufgehoben wird noch für die falschen Server gilt: Das Fehlerpotenzial einer solchen Lösung ist hoch.

Besser ist es, eine Zeitzone zu wählen, die keiner Umstellung oder Anpassung unterliegt. Hierfür bietet sich die UTC-Zeitzone an.

UMSETZUNG AM MPI FÜR BIOPHYSIKALISCHE CHEMIE

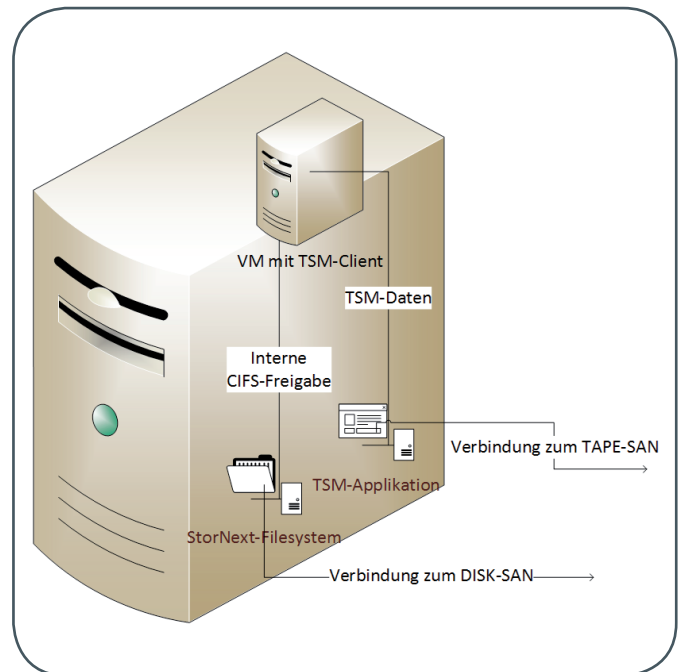
Das MPI-BPC hat für die Sicherung des StorNext-Filesystems einen eigenen Server beschafft. Die Konfiguration entspricht damit der in Abb. 1 dargestellten. Obwohl der Server Mitglied in der Instituts-AD-Domäne ist, konnte er problemlos auf die UTC-Zeitzone umgestellt werden. Der Vergleich mit den Fileservern zeigt, dass alle Dateien auf dem TSM-Server tatsächlich eine oder zwei Stunden Abweichung in der Änderungszeit aufweisen, die UTC-Zeitstempel aber – wie erwartet – sich zur Zeitumstellung nicht geändert haben.

UMSETZUNG AM MPI FÜR EXPERIMENTELLE MEDIZIN

Am MPI EM gibt es im StorNext-Filesystem einen Server, der nur wenige Clients bedient und daher für die Nutzung als TSM-Server ausreichend Ressourcen besitzt. Allerdings müssen alle Gateway-Server, die Netzwerkfreigaben bereitstellen, zur Vereinfachung der Administration nahezu identisch konfiguriert sein, insbesondere die Zeitzone muss zu den übrigen passen. Einen weiteren Server zu beschaffen und diesen in das SNFS zu integrieren, erhöht den Komplexitätsgrad der SNFS-Konfiguration. Außerdem stellt sich die Frage nach der Wirtschaftlichkeit. Dieser Gateway-Server sollte daher als TSM/ISP-Server fungieren und es musste ein geeigneter Ansatz gefunden werden, das SNFS sowohl performant wie mit – vergleichsweise – geringer zusätzlicher Komplexität zu sichern.

Vom MPI EM kam der Vorschlag, eine VM (Virtual Machine) zu erstellen, die den TSM/ISP-Client ausführt, aber durch die direkte Kommunikation über ein „Host-Only“-Netzwerk keinen „echten“ Netzwerk-Traffic erzeugt und durch die direkte Koppelung auch ausreichend leistungsfähig ist. Es muss lediglich darauf geachtet werden, keinen SMB-Multichannel zu konfigurieren, da dieser bei vorhanden 1GE-Netzwerkkarten von diesen in seiner Leistungsfähigkeit begrenzt wird. Dieser Ansatz benötigt keine oder nur wenige zusätzliche Ressourcen (ggf. weiteren Hauptspeicher).

Es gab anfänglich auch Überlegungen, TSM/ISP vollständig in die VM zu verlegen. Hierzu hätten aber sämtliche SAN-Verbindungen, also neben den Volumes für das StorNext-Filesystem



2_Schematische Darstellung der Lösung am MPI EM mit einer VM für den TSM/ISP-Client

auch die Anbindungen an die Tape-Library, zur VM durchgereicht werden müssen. Dies ist zwar mit der Microsoft-Virtualisierung „Hyper-V“ möglich, allerdings hätten sowohl der Host-Server als auch wahrscheinlich das komplette Filesystem-Setup neu konfiguriert werden müssen. Dieser Ansatz wurde daher als unpraktisch verworfen.

Umgesetzt wurde der ursprüngliche Ansatz, ausschließlich den TSM/ISP-Client in die VM zu bringen (siehe Abb. 2). Dies bietet zahlreiche Vorteile: Die SAN-Verbindungen müssen nur zum Host, nicht aber durch die Virtualisierung konfiguriert werden, letztendlich also nur die Tape-Laufwerke hinzugefügt werden. Der Zugriff auf das SNFS erfolgt über eine CIFS-Freigabe zur VM. Der Rückweg der Daten nutzt zwar eine TCP/IP-Verbindung, diese liegt aber vollständig innerhalb des Hosts und kann entsprechend beliebig und performant konfiguriert werden. In dieser Konfiguration läuft der TSM/ISP-Server weiter direkt auf dem Host, es sind also hier keine Einbußen durch die Virtualisierung zu erwarten.

EXKURS: WORKAROUND FÜR VIRTUALMOUNT-POINTS BEI WINDOWS-CLIENTS

Für UNIX und Linux sowie macOS gibt es im TSM/ISP die Möglichkeit, einzelne Verzeichnisse als virtuelle Laufwerke im TSM zu konfigurieren. Hiermit vereinfacht sich die Konfiguration des Backups, da direkt das virtuelle Laufwerk als Backup-Quelle angegeben werden kann, anstatt das tatsächliche Laufwerk anzugeben und alle Verzeichnisse, die nicht gesichert werden sollen, mittels Exclude-Regeln auszuschließen.

Eine vergleichbare Funktion gibt es für Windows leider nicht. Damit entfällt auch die Möglichkeit, das Backup über verschiedene virtuelle Laufwerke zu parallelisieren.

Sofern aber nur einzelne Verzeichnisse gesichert werden sollen, nicht aber auch parallel das „übrige“ Wurzelverzeichnis, müssen meist zahlreiche Ausschlussregeln in Form von *exclude.dir*-Einträgen in der *dsm.opt* erstellt werden. Selbstredend ist dieser Weg hochgradig fehlerträchtig, zusätzlich werden natürlich alle

im Wurzelverzeichnis eines Laufwerks neu erstellten Verzeichnisse nicht automatisch ausgeschlossen, sondern in das Backup hineingenommen.

Mit dem folgenden Workaround kann unter Windows die Konfiguration vereinfacht und die Parallelisierung des Backups erreicht werden:

- Man erstelle für jedes Verzeichnis, das gesichert werden soll, eine „erweiterte Freigabe“.
- Durch das Hinzufügen eines \$ an den Freigabenamen wird dieser vom Windows-SMB-Dienst auch nicht in der Netzwerkübersicht aufgelistet („versteckte Freigabe“).
- Zugriff auf diese Freigabe benötigen nur die lokalen Administratoren des Backup-Knotens.
- Die zu sichernden Pfade können über das Loopback-Device angesprochen werden:
`DOMAIN \127.0.0.1<Freigabe1>`
`DOMAIN \127.0.0.1<Freigabe2>`

Aus Sicht des TSM/ISP-Clients sind die Freigaben unabhängige Netzwerkshares und können parallel gesichert werden!

ERFAHRUNGEN

Durch die Umstellung des MPI-BPC-Servers auf UTC und die Einführung des Backups über Netzwerkfreigaben im Januar 2018 liegen für diesen die meisten Erfahrungswerte sowohl allgemein als auch im Vergleich zu den Vollsicherungen zur Zeitumstellung aus den Vorjahren vor:

- **Datendurchsatz:** Im Rahmen der Neusicherung mussten alle Datenbereiche nochmals als Vollbackup gesichert werden. Hierbei wurden einzelne Freigaben mit einem Durchsatz von bis zu 40 TB/Tag gesichert. Übliche Werte bei der LAN-Sicherung liegen im Bereich von 5 – 10 TB/Tag. Auch bei den täglichen Sicherungen lassen sich Durchsatzraten von bis zu 1,6 GB/s bei einzelnen Knoten beobachten – parallel zu weiteren laufenden Sicherungen. Zum Vergleich: Beim TSM-Server SM131 (Uni) erscheinen die Zahlen zunächst ähnlich (maximal 50 parallele Sessions, bis zu 1,8 GB/s). Allerdings sind die übertragenen Datenmengen bei den schnellsten Sessions erheblich geringer (wenige Megabyte statt mehrere Gigabyte oder Terabyte). Sessions mit ähnlichen Datenmengen liegen bei deutlich geringeren (effektiven) Durchsätzen.
- **Lookup-Time:** Obwohl über die Nacht teilweise fast 40 Knoten gleichzeitig auf dem MPI-BPC-Server sichern, werden pro Sicherungsprozess bis zu 6,2 Mio. Objekte pro Stunde durchsucht, im Mittel immer noch rund 1,6 Mio. Objekte pro Stunde: Der mittlere Durchsatz ist zwar mit den Zahlen eines einzelnen Knotens vergleichbar (1 – 2 Mio. Objekte pro Stunde), aber es sind sowohl der Spitzenwert als auch die Gesamtleistung deutlich höher, da

diese bei Berücksichtigung der Parallelisierung die Summe von bis zu 40 „durchschnittlichen“ Sicherungen umfasst.

- Die Umstellung auf die (lokalen) Freigaben führt zu keinen bzw. nicht messbaren Änderungen der Leistungsfähigkeit.

Beim Server des MPI EM kommt als zusätzliche Komponente die Virtualisierung des TSM/ISP-Clients ins Spiel. Da diese kurz nach der Inbetriebnahme erfolgte, können hier vor allem die „Kosten“ hierfür betrachtet werden:

- Während bei der Konstellation ähnlich dem MPI-BPC (Client direkt auf dem Server) zunächst ebenfalls Durchsatzraten von bis zu 1,5 GB/s zu beobachten waren, liegen sie nun in der Regel bei etwa 300 MB/s – allerdings sind die zu sichernden Datenmengen in diesen Fällen auch erheblich geringer. Fallen große inkrementelle Sicherungen an, so liegt der Durchsatz bei bis zu 1 GB/s.
- Insgesamt kommt es durch die Virtualisierung zu Einbußen im Bereich von etwa 30 % – allerdings auf hohem Niveau!

BEWERTUNG

Abschließend ist festzustellen, dass mit dem Ansatz, einen TSM/ISP-Server in ein globales Filesystem zu integrieren, erhebliche Vorteile insbesondere im Hinblick auf die Backup-Leistung zu erreichen sind. Zumindest im Umfeld der GWDC sind die globalen Filesysteme sowieso so groß und enthalten so viele Files (Objekte), dass ein dedizierter Server Sinn macht.

Das Problem der Zeitumstellung beim StorNext-Filesystem konnte mit dem Einstellen der UTC-Zeitzone für den Client gelöst werden.

Der im Exkurs beschriebene Workaround für die fehlende *VIRTUALMOINT*-Option erfüllt alle Erwartungen. Der zusätzliche Konfigurationsaufwand bei Windows wird durch die wesentlich vereinfachte TSM/ISP-Konfiguration aufgewogen, zudem sind keine Leistungseinbußen feststellbar.

AUSBLICK

Auch wenn die zentralen Windows-Filesysteme der GWDC vorrangig über ein NetApp-Cluster bereitgestellt werden, so verbleiben die „Spezial-Filesysteme“ doch weiterhin im StorNext und müssen dort gesichert werden. Auch für die NetApp-Filer soll neben den naheliegenden Sicherungsansätzen (Snapshots/Schattenkopien, Replikation) eine Katastrophenfall-Vorsorge mit Applikations- und Medienbruch (Einsatz einer unabhängigen Software zur Sicherung auf eine andere Art von Datenträgern) umgesetzt werden. Der hierfür genutzte TSM-Server wird für die Sicherung der „Spezial-Filesysteme“ wieder ins SNFS integriert werden, um die dortigen großen Datenmengen effizient sichern zu können. ■

Continuous Delivery für das Kundenportal der GWGD

Text und Kontakt:

Christof Pohl
christof.pohl@gwdg.de
0551 201-1878

Continuous Delivery und *Infrastructure as Code* versprechen hohe Qualitätsstandards in der Softwareentwicklung und transparente Kommunikation von Fortschritten gegenüber Auftraggebern und Partnern von Entwicklungsprojekten. Bei der GWGD wurde eine solche Lösung auf Basis von Open-Source-Software am Beispiel des Kundenportals realisiert.

EINLEITUNG

Die GWGD ist als Kompetenzpartner für IT im Kontext nationaler und internationaler Forschungsprojekte [1] sowie als Hochschulrechenzentrum und IT-Dienstleister der Georg-August-Universität Göttingen und der Max-Planck-Gesellschaft (MPG) an zahlreichen Softwareentwicklungsprojekten beteiligt. Im Interesse der jeweiligen Partner strebt die GWGD dabei durch den Einsatz *agiler Methoden* [FH01] in der Entwicklung eine hohe Transparenz, verlässliche Zeit- und Kostenschätzungen, die Berücksichtigung fluktuierender Anforderungen sowie die Gewährleistung einer hohen Produktqualität an [vgl. Ver14, S. 7–9]. Aus dem Kanon der agilen Methoden schließt *Continuous Delivery* die Lücke zwischen Entwicklung und Betrieb und leistet somit einen elementaren Beitrag zur Erreichung der genannten Ziele [vgl. HF10, S. 10–13]. Dieser Artikel stellt beispielhaft die Umsetzung von *Continuous Delivery* für das Kundenportal der GWGD sowie die dafür verwendeten Konzepte und technischen Lösungen vor.

DAS KUNDENPORTAL DER GWGD ALS ANWENDUNGSFALL FÜR CONTINUOUS DELIVERY

Das Kundenportal der GWGD [2] (siehe Abb. 1) dient im Web als zentrale Anlaufstelle für Informationen und Selfservices zu den angebotenen IT-Dienstleistungen. Technisch basiert das Kundenportal auf der Open-Source-Web-Plattform *Liferay Portal Community Edition (CE)* in Version 6.2. Dabei fungiert Liferay u. a. als Entwicklungs- und Integrationsplattform für die Selfservice-Funktionen. Bei letzteren handelt es sich um von der GWGD entwickelte *Liferay-Plugins*, auch *Portlets* genannt, die i. Allg. einfache, webbasierte GUIs [3] für die Beantragung und Konfiguration von GWGD-Diensten bereitstellen.

Als Laufzeitumgebung für Liferay werden bei der GWGD das *Oracle Java Development Kit (JDK)* und der *GlassFish JEE3 Application Server* [4] genutzt. Die Portlets mit den Selfservice-Funktionen sind als JEE-Anwendungen realisiert und nutzen bei Bedarf *MySQL-Datenbanken*. Als Build-Tool für die Übersetzung der Quelltexte und die Ausführung von Unit-Tests kommt *Apache Maven*



Continuous Delivery for the GWGD Customer Portal

The GWGD functions as an IT competence partner in national and international research projects [1] as well as a data center and IT service provider for the Georg-August-Universität Göttingen and the Max Planck Society. In these contexts it is also participating in numerous software development projects. In behalf of the respective partners, GWGD strives for high transparency, reliable time and cost estimates, quick responses to changing requirements and high-quality software products through the use of agile methodologies in software development. [see Ver14, pp. 7–9]. From the group of agile methodologies, *Continuous Delivery* closes the gap between development and operation and thus makes a fundamental contribution to achieving the aforementioned goals. This article presents a *Continuous Delivery* solution for the GWGD Customer Portal as well as concepts and technical implementation behind it.

1_Startseite des Kundenportals der GWGD

zum Einsatz. Einstellungen für den Zugriff auf Webservice-APIs [5] der als Selfservice eingebundenen GWDG-Dienste, z. B. URLs und Zugangsdaten, sind als Systemvariablen in der Konfiguration des GlassFish-Servers hinterlegt.

Dementsprechend kompliziert und zeitaufwendig stellen sich die Bereitstellung aktueller Softwareversionen und Serverinstanzen sowohl in der Staging- [6] als auch in der Produktivumgebung des Kundenportals dar. Updates werden zumeist nicht umgehend vorgenommen, sondern gesammelt und auf einen späteren Zeitpunkt verschoben. Dieses Verhalten sorgt dafür, dass die Komplexität von Wartungsarbeiten ansteigt, und es fördert eine „If it ain't broke, don't fix it“-Mentalität der Administratoren. Hinsichtlich aktueller Entwicklungen, z. B. zur Integration neuer Selfservice-Funktionen, kann somit die Transparenz gegenüber den Stakeholdern des Kundenportal-Projekts nicht in dem gewünschten Umfang gewährleistet werden. Hierfür musste eine passende Lösung gefunden werden.

CONTINUOUS-PARADIGMEN IN DER SOFTWAREENTWICKLUNG

Drei Continuous-Paradigmen adressieren die o. g. Herausforderungen durch den Einsatz verschiedener Methoden und Praktiken zur kontinuierlichen Bereitstellung von Anwendungen und IT-Diensten.

Continuous Integration

Die Methoden von *Continuous Integration (CI)* haben zum Ziel, die Zusammenführung parallel laufender Arbeiten innerhalb eines Softwareentwicklungsprojektes zu unterstützen und zu automatisieren. Hierbei kommen i. Allg. ein *Software Configuration Management (SCM)*, beispielsweise *Git* [7] oder *Apache Subversion* [8], sowie ein *Datenrepositorium* wie der *Sonatype Nexus Repository Manager (NXRM)* [9] oder *Apache Archiva* zum Einsatz. Durch jeden Commit eines Entwicklers in das SCM-System wird ein sogenannter *Job* ausgelöst, der zunächst die aktualisierten Quelltexte übersetzt und vorhandene Softwaretests, zumeist Unit- oder Integrationstests, durchführt. Genügen die Ergebnisse der Softwaretests den vorgegebenen Qualitätskriterien (werden also z. B. bei ausreichender Testabdeckung der Quelltexte alle Tests erfolgreich absolviert), wird ein neues Softwarepaket erstellt und im Datenrepositorium abgelegt.

Darüber hinaus werden mit dem Einsatz von CI auch organisatorische Aspekte verbunden, u. a. die Verwendung von Branches und Forks im SCM-System zur Strukturierung der Entwicklungsarbeiten oder die umgehende Behebung von Fehlern bei der Übersetzung der Software.

Die gewünschte Erhöhung der Softwarequalität und die Reduzierung des Aufwands für Fehlersuche und -behebung werden im Kontext von CI also hauptsächlich durch die Standardisierung der Übersetzungs- und Testumgebung und die umgehende Rückmeldung von Fehlern an die Entwickler erreicht.

Continuous Delivery

Continuous Delivery (CD) umfasst die Methoden von CI und erweitert dessen Einsatzbereich auf alle Folgeschritte, die zur Bereitstellung der Software in einer Laufzeitumgebung notwendig sind [vgl. HF10, S. 106 ff.]. Zielsetzung ist dabei, die durch den Einsatz von CI immer in einer aktuellen Version verfügbare Software

automatisiert in der Staging-Umgebung installieren zu können. Dieser Ablauf wird in Form von *Pipelines* umgesetzt, die mehr oder weniger komplexe und in sich abgeschlossene Aktivitäten bzw. Operationen zur Automatisierung der Installation umfassen. Unterstützt wird CD durch eine *Konfigurationsverwaltung* (häufig ebenfalls ein SCM-System), die zur Speicherung sogenannter *Configuration Items*, z. B. Konfigurationsdateien oder Automatisierungsskripts, dient. CD ermöglicht somit eine frühzeitige und häufige Bereitstellung von Beta-Versionen der Software, um so die aktuellen Entwicklungsarbeiten für Auftraggeber bzw. Projektpartner transparent zu machen.

Der Einsatz von CD kann sinnvoll mit weiteren Methoden aus dem Umfeld von *DevOps* und *Infrastructure as Code* kombiniert werden. Das DevOps-Paradigma propagiert dabei eine Abkehr von der bis dato oftmals praktizierten Trennung zwischen Entwicklung (*Development*) und Betrieb (*Operating*) hin zu einer gemeinschaftlichen Kultur zur Bereitstellung von IT-Dienstleistungen (*DevOps*). Ursprünglich wurde das DevOps-Paradigma im Kontext des Cloud-Computings, insbesondere *Infrastructure as a Service (IaaS)*, populär, durch das u. a. die horizontale Skalierung von IT-Infrastrukturen, insbesondere im Bereich der Server, um Größenordnungen gesteigert wurde. Die manuelle Administration hunderter oder gar tausender Server, die gemeinsam einen Dienst bereitstellen, führt allerdings zwangsläufig zu Inkonsistenzen und damit potenziell auch zu Beeinträchtigungen der Dienstqualität. [vgl. Lou12]

Ein Lösungsansatz dafür ist *Infrastructure as Code*, welcher durch Maßnahmen zur Automatisierung in Kombination mit einem SCM-System die Gewährleistung von Konsistenz und Reproduzierbarkeit bei der Installation und Administration großer IT-Infrastrukturen verspricht. Die Automatisierung erfolgt dabei mittels geeigneter Tools wie *Puppet* [10], *Ansible* [11] oder *Chef* [12] und sorgt dafür, dass sich das System stets auf einem aktuellen Stand befindet.

Obwohl DevOps und Infrastructure as Code ihren Ursprung im Cloud-Computing haben, eignen sie sich auch im Kontext *Serviceorientierter Architekturen* oder *Microservices* dafür, die Qualität von IT-Diensten durch die Automatisierung komplexer Server-Setups und Anwendungs- bzw. Schnittstellenkonfigurationen zu erhöhen. Dies gilt auch für das GWDG-Kundenportal, das im weiteren Verlauf stellvertretend für solche IT-Dienste betrachtet wird.

Continuous Deployment

Obwohl Continuous Delivery grundsätzlich alle notwendigen technischen Voraussetzungen für eine automatische Installation der Software im Produktivbetrieb abdeckt, wird es heute zumeist als Paradigma für den Staging-Bereich verstanden. Dementsprechend befasst sich *Continuous Deployment* [13] hauptsächlich mit der automatischen Installation einer Anwendung im Produktivbetrieb und ist somit als Erweiterung von Continuous Delivery zu verstehen. Durch den Einsatz von Continuous Deployment profitieren sowohl Auftraggeber und Projektpartner als auch Nutzer einer Anwendung von kurzfristigen Fehlerbehebungen und häufigeren, dafür aber zumeist kleineren Funktionserweiterungen [vgl. Fow06, Abschnitt „Benefits of Continuous Integration“].

Die technische Umsetzung erfolgt analog zu CD, sodass sich hier i. Allg. keine zusätzlichen Herausforderungen stellen. Gegebenenfalls ist aber zur produktiven Inbetriebnahme einer neuen Softwareversion im Rahmen eines IT-Service-Managements wie der *IT Infrastructure Library* [AXE] oder zur Erfüllung von

Qualitätsmanagementnormen wie *DIN EN ISO 9001* [DIN15] die Einhaltung bestimmter Konventionen, z. B. eine vorherige Schulung des *Service Desks*, zu gewährleisten. Aber auch informelle Gründe, etwa ein gefühlter Verlust an Kontrolle über das angebotene Endprodukt oder der Wunsch, dass neue Releases ein Mindestmaß an Innovation enthalten, können im Kontext von CD für manuelle Installationen im Produktivbetrieb sprechen.

ANFORDERUNGEN AN DEN EINSATZ VON CONTINUOUS DELIVERY FÜR DAS GWDG-KUNDENPORTAL

Eine CD-Lösung für das Kundenportal kann auf die dort bereits vorhandenen und mit Maven automatisierten Software-Builds und -tests aufsetzen. Dies gilt in ähnlicher Form für einen großen Teil der Entwicklungsprojekte der GWDG, wobei sich die jeweiligen CI-Methoden in Umfang sowie Art und Weise der Umsetzung unterscheiden. Demzufolge wird der CI-Bereich im Rahmen dieses Artikels nicht im Detail behandelt.

Staging-Umgebung

Ganz im Sinne der agilen Vorgehensmodelle in der Softwareentwicklung ist eine der wichtigsten Anforderungen an die zu entwickelnde CD-Lösung, dass Quelltextänderungen möglichst kurzfristig in Form entsprechend aktualisierter Kundenportal-Versionen auf dem Staging-Server bereitgestellt werden. Gleiches gilt für Änderungen in der Konfigurationsverwaltung von Server-Betriebssystem und Technologiestack für das Kundenportal. Dazu ist insbesondere die Installation von Laufzeitumgebung und Kundenportal-Software auf dem Server zu automatisieren. Durch diese Maßnahme soll eine möglichst geringe Zyklusdauer für die Verfügbarkeit von Änderungen aus der Kundenportal-Entwicklung im Staging-Betrieb erreicht werden.

Ein weiteres Ziel ist eine möglichst hohe Übereinstimmung der Betriebsparameter aller Systemkomponenten zwischen Staging- und Produktivserver. Dies umfasst insbesondere die Versionen und Konfigurationen des Betriebssystems, der verwendeten Serverdienste sowie des Technologiestacks für das Kundenportal. Eine Ausnahme bilden geplante Updates oder Konfigurationsänderungen; hierfür ist es explizit erwünscht, dass diese vor der produktiven Inbetriebnahme ausreichend in der Staging-Umgebung getestet werden können.

Weiterhin ist die Verwendung der *GWDG Cloud Server* [14] als IaaS-Plattform für den Staging-Betrieb vorgesehen. Hierdurch soll die schnelle und bedarfsorientierte Bereitstellung einer oder mehrerer Server-Instanzen ermöglicht und somit eine höhere Flexibilität als bei der Verwendung von Hardware-Servern oder dedizierten VMs [15] geboten werden.

Entwickler-VMs

Im Rahmen der Kundenportal-Entwicklung wird bereits das Tool *Vagrant* [16] zur Installation von lokalen Kundenportal-VMs auf den Workstations der Entwickler verwendet. Als Virtualisierungsplattform kommt *VirtualBox* [17] zum Einsatz. Hierfür werden Datenbank-Dumps und ein Archiv der Liferay-Arbeitsdaten derzeit noch in unregelmäßigen Abständen manuell vom jeweiligen Stand des produktiven Kundenportal-Servers erstellt. Gleiches gilt für die sogenannte *Vagrant-Box*, ein *Virtual Machine Image (VMI)* mit einer Basisinstallation des Betriebssystems, und

die Provisionierungsskripts, in denen u. a. die zu installierenden Dienste sowie Konfigurationseinstellungen für den Server festgelegt werden.

Aufgrund der grundsätzlich positiven Erfahrungen mit *Vagrant* und *VirtualBox* sollen diese weiterhin für die Bereitstellung von Entwickler-VMs zum Einsatz kommen; gleiches gilt für die Verwendung von *Git* zur Verwaltung der Provisionierungsskripts. Größere Dateien (*Vagrant-Box*, Archiv der Liferay-Arbeitsdaten, Datenbank-Dumps etc.) hingegen sollen in einem dedizierten Datenrepositorium abgelegt werden. Zudem sollen diese im Zuge des CD-Prozesses regelmäßig und automatisiert erstellt bzw. aktualisiert werden. Nicht zuletzt sollen Entwickler-VMs nicht mehr nur lokal in *VirtualBox*, sondern wahlweise auch in Form eines GWDG Cloud Servers instanziiert werden können.

Ansonsten werden bei den Entwickler-VMs vergleichbare Anforderungen bezüglich der kurzfristigen Verfügbarkeit und Aktualität von Kundenportal-Software und Konfigurationen wie im Kontext des Staging-Servers gestellt.

CINDY4G: CONTINUOUS DELIVERY FÜR DIE GWDG

Im Rahmen eines Projektes zur Umsetzung von CD für das Kundenportal wurden zunächst ein generisches Modell sowie eine Systemarchitektur für *Cindy4G (Continuous Delivery für die GWDG)* entworfen. Dabei wurden Modell und Architektur nicht spezifisch auf das GWDG-Kundenportal ausgelegt, sondern lassen sich grundsätzlich für beliebige Anwendungen und IT-Dienste und somit auch in Kontexten der MPG und der Universität Göttingen verwenden.

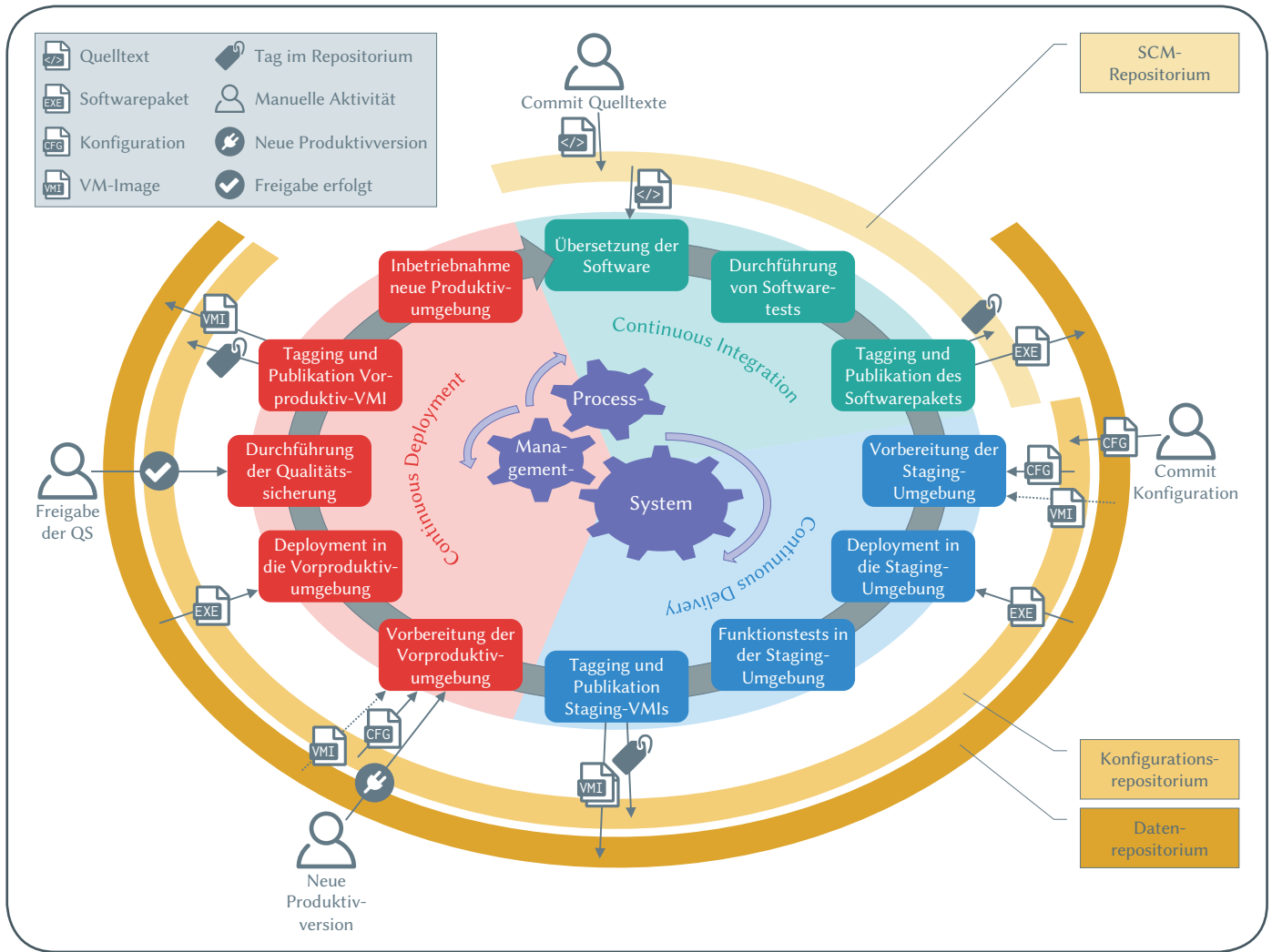
Als Basis für das generische Modell wurden die *Deployment Production Line* von Humble, Read und North [HRN06] sowie exemplarische Referenzarchitekturen renommierter Firmen aus der IT-Branche wie *IBM* [18] [Zhu14] und *CloudBees* [19] [Le 14] herangezogen.

Im Mittelpunkt des generischen Modells und der Systemarchitektur stehen der CD-Prozess sowie die Systemkomponenten für dessen Umsetzung; außerdem wird die Interaktion der verwendeten Komponenten samt Ein- und Ausgabeartefakten dargestellt. Aufgrund der großen inhaltlichen Nähe umfasst das *Cindy4G*-Konzept neben CI und CD auch die zusätzlichen Schritte für Continuous Deployment. Hierdurch soll die Möglichkeit einer dahin gehenden, späteren Erweiterung von *Cindy4G* sichergestellt werden.

Generisches Modell

Der übergeordnete *CD-Prozess* des generische *Cindy4G*-Modells (siehe Abb. 2) ist in die Phasen Continuous Integration (grün), Continuous Delivery (blau) und Continuous Deployment (rot) aufgeteilt. Diese umfassen ihrerseits jeweils eine Abfolge verschiedener Pipelines, in der Abbildung als Kästchen in der jeweiligen Farbe dargestellt. Die Steuerung der Prozesse und Pipelines erfolgt durch ein zentrales *Process-Management-System*. Zur Unterstützung des CD-Prozesses sind drei Repositorien vorgesehen: ein *SCM-Repository* (als Teil eines SCM-Systems) für die Softwarequelltexte, ein *Konfigurationsrepositorium* (als Teil einer Konfigurationsverwaltung) zur Ablage von Configuration Items für die Staging- und Produktivumgebung sowie ein *Datenrepositorium* zur Speicherung von Artefakten wie Softwarepaketen und VMIs.

Abb. 2 zeigt entlang des grauen Pfeils den generischen



2_Cindy4G – Generisches Modell

CD-Prozess von Cindy4G sowie die Interaktion mit den verschiedenen Repositorien samt Ein- und Ausgabeartefakten für die jeweiligen Pipelines. Der übliche Ablauf beginnt im oberen Teil des Diagramms mit einem Quelltext-Commit für die Anwendung, woraufhin die verschiedenen CI-Pipelines automatisch abgearbeitet werden. Das Ergebnis der CI-Phase ist dann ein neu erstelltes Softwarepaket, welches im Datenrepositorium abgelegt wird. Danach startet automatisch die CD-Phase, in der zunächst unter Verwendung der Configuration Items aus der Konfigurationsverwaltung eine aktuelle Staging-Umgebung vorbereitet und anschließend das neue Softwarepaket installiert wird. Zur weiteren Verwendung u. a. als Entwickler-VMs wird im letzten Schritt ein VMI der neuen Staging-Umgebung angefertigt und im Datenrepositorium bereitgestellt.

Das generische Modell sieht in der anschließenden Continuous-Deployment-Phase die Möglichkeit vor, eine automatische Installation der neuen Softwareversion in der Produktivumgebung durchzuführen. Dieser Teil wird im Falle des Kundenportals aufgrund der an früherer Stelle genannten Einwände allerdings weiterhin manuell ausgeführt.

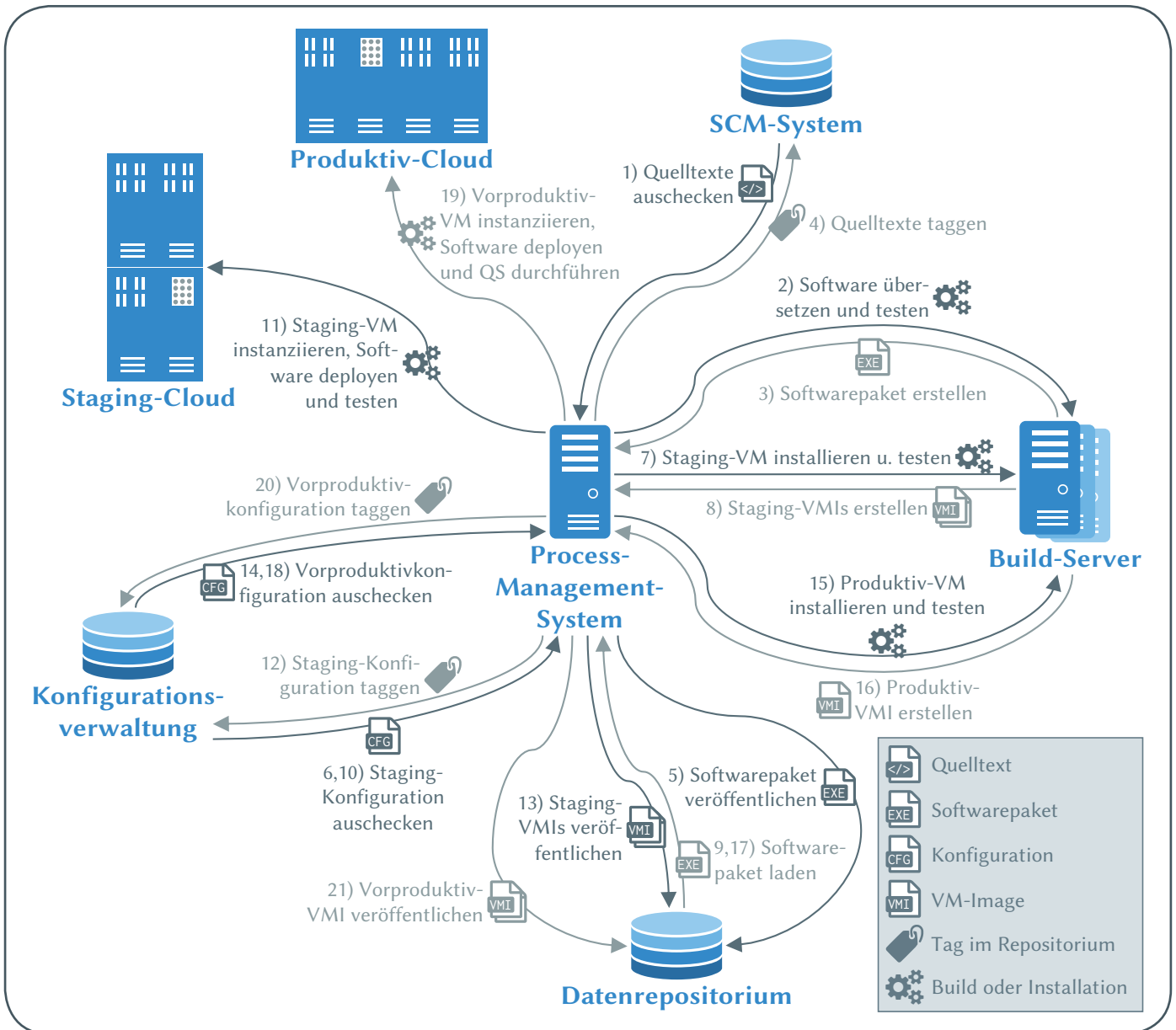
Nicht explizit im Modell dargestellt ist die Behandlung ggf. auftretender Fehler. Anhand gängiger *Best Practices* aus dem CI- bzw. CD-Umfeld sollte ein Fehler in jedem Fall zum Abbruch des laufenden CD-Prozesses führen und Entwickler bzw. Dienstbetreiber z. B. per E-Mail benachrichtigt werden. Dies ermöglicht eine

zeitnahe und damit in vielen Fällen vergleichsweise unkomplizierte Behebung des Fehlers, da diese in den meisten Fällen durch kürzlich vorgenommene Änderungen hervorgerufen werden und diese bei den jeweiligen Mitarbeitern noch präsent sind. Die aufwendige Einarbeitung in bereits länger zurückliegende Code-Änderungen kann somit oftmals entfallen. Ist der Fehler behoben, sollte der CD-Prozess idealerweise mit dem zuletzt fehlgeschlagenen Schritt fortgesetzt werden können, um die ggf. langwierige Ausführung vorher bereits erfolgreich durchlaufener Pipelines zu vermeiden.

Systemarchitektur

Die Systemarchitektur von *Cindy4G* setzt sich aus den bereits genannten Systemen (Process Management, SCM, Konfigurationsverwaltung und Datenrepositorium) sowie einem oder mehreren *Build-Servern* und *IaaS-Clouds* für die Staging- und Produktivumgebung zusammen. Abb. 3 zeigt darüber hinaus die Interaktionen, die im Zuge der Durchführung des CD-Prozesses zwischen den beteiligten Systemen stattfinden.

Das *Process-Management-System* dient zur Automatisierung des CD-Prozesses und zur Orchestrierung aller Systemkomponenten. Dazu wird die Möglichkeit zur Ausführung einer breiten Palette von Tools, z. B. für die Automatisierung des Software-Builds oder die Installation und Konfiguration von VMIs, ebenso vorausgesetzt wie die Unterstützung gängiger Schnittstellentechnologien zur Kommunikation mit den anderen Komponenten. Um die



3_Cindy4G – Systemarchitektur

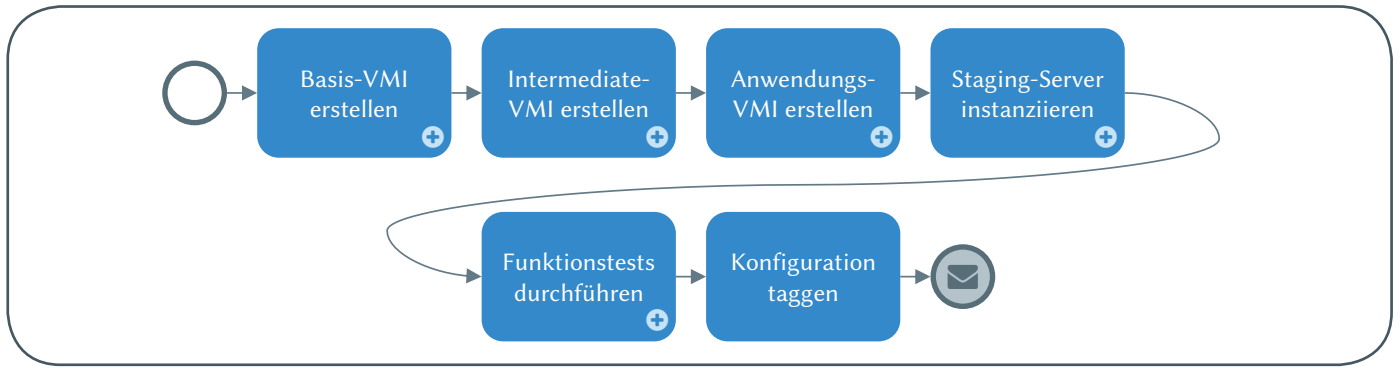
Aktualität aller erzeugten Artefakte sowie der Staging- bzw. Produktivumgebung sicherzustellen, ist eine ereignis- und zeitgesteuerte Initiierung des Prozesses vorgesehen, z. B. nach jedem SCM-Commit oder einmal täglich als Nightly Build.

Im *SCM-System* und der *Konfigurationsverwaltung* werden hauptsächlich Textdateien abgelegt, sodass i. Allg. ein gemeinsames System, z. B. *Git* oder *Subversion*, für beide Zwecke verwendet werden kann. Im *Datenrepositorium* werden hingegen größtenteils Binärdateien gespeichert, wobei die Größe einzelner Dateien z. B. im Falle von VMIs auch mehrere Gigabyte betragen kann. Hierfür sind Systeme wie *Git* und *Subversion* weniger geeignet, sodass ein auf Binärdateien spezialisiertes Repository wie *NXRM* oder *Archiva* verwendet werden sollte.

Staging- und Produktiv-Cloud (sofern auch Continuous Deployment umgesetzt werden soll) können hingegen ebenfalls auf dem gleichen IaaS-Dienst basieren. Dies ermöglicht die Verwendung einheitlicher Implementierungen für die Cloud-API. Womöglich sprechen allerdings unterschiedliche Anforderungen im Hinblick auf Verfügbarkeit, Datenschutz und -sicherheit sowie Betriebskosten dafür, verschiedene IaaS-Dienste mit jeweils passenden

Service Level Agreements (SLAs) für die Staging- und Produktivumgebung zu verwenden.

Der *Build-Server* ist für das Übersetzen und Testen der Software sowie für die Installation und Konfiguration von VMIs zur späteren Verwendung in der Entwicklung oder in der Staging- bzw. Produktivumgebung vorgesehen. Dafür sind die entsprechenden Tools bereitzustellen, z. B. Build- und Test-Tools für die Software sowie Virtualisierungsplattform und Tools zur automatischen Installation von VMs. Bei umfangreichen Anwendungen können die Übersetzung der Quelltexte und die anschließende Durchführung von Unit- und Integrationstests hohe Anforderungen an die Hardwareressourcen des Build-Servers stellen. Gleiches gilt für die Erstellung von VMIs, da während der Installation und Konfiguration eine bezüglich ihrer virtuellen Ressourcen passend dimensionierte VM auf dem Build-Server ausgeführt werden muss. Aus diesem Grund erscheint es sinnvoll, bei Bedarf mehrere Build-Server bereitzustellen, auf die anstehende Tasks verteilt werden können. Hier ist insbesondere das *Process-Management-System* gefordert, eine einfache und effiziente Lastverteilung zwischen den Build-Servern zu ermöglichen.



4_Cindy4G – Mehrstufiger VMI-Build

Mehrstufige Pipeline für VMI-Builds

Anhand des generischen Modells und der Interaktionen innerhalb der Systemarchitektur ist erkennbar, dass Cindy4G die Bereitstellung konsistenter Laufzeitumgebungen für die Staging- und Produktivumgebung durch die Erzeugung und Instanziierung verschiedener VMIs sicherstellt. Die VMIs enthalten dabei das Betriebssystem mit den jeweiligen Konfigurationsanpassungen sowie den für die Software bzw. den Dienst benötigten Technologiestack.

Da die Installation von VMs sowie die anschließende Erzeugung entsprechender VMIs zeit- und ressourcenaufwendig sind, wird ein mehrstufiger VMI-Build für eine bessere Wiederverwendbarkeit der Images vorgesehen. Ein solches Konzept wurde bereits von Google in [Goo17] beschrieben und kann für Cindy4G in leicht abgewandelter Form verwendet werden. Konkret wird die Image-Erstellung in die Stufen Basis-, Intermediate- und Anwendungs-VMI unterteilt. Abb. 4 zeigt die entsprechend angepasste CD-Phase aus dem generischen Cindy4G-Modell, wobei die Ablage der Images im Datenrepositorium Teil der jeweiligen VMI-Builds ist.

Das Basis-VMI: Die Erstellung des Basis-VMIs umfasst im Wesentlichen die Installation und Konfiguration des Betriebssystems. Je nach Distribution stehen hierfür verschiedene Tools zur Verfügung, für Debian und Ubuntu z. B. der *Debian Installer/Preseed* [20]. Um das Basis-VMI nach Möglichkeit in verschiedenen CD-Prozessen einsetzen zu können, sollten installierte Linux-Pakete sowie Änderungen an Konfigurationsdateien von allgemeiner Natur und möglichst unabhängig von späteren Einsatzzwecken der VM sein.

Das Intermediate-VMI: Unter Verwendung des Basis-VMIs wird danach eine Intermediate-VM erstellt, die hauptsächlich den Technologiestack für die spätere Anwendung enthält. Dabei sollte das Intermediate-VMI so installiert und konfiguriert werden, dass es für verschiedene Anwendungen auf Basis gleichartiger Technologiestacks verwendbar ist. Als Beispiele mit Bezug zum Kundenportal seien der *Göttingen Campus Event Calendar* [21], die *Academic Cloud* für die niedersächsischen Hochschulen [22] sowie die *Virtuelle Forschungsumgebung soeb 3* [23] genannt, die ebenfalls Liferay Portal und JEE-Technologien verwenden. Aber auch im Kontext eines einzelnen CD-Prozesses ist die Verwendung eines Intermediate-VMIs sinnvoll. So kann z. B. nach einem Commit ins SCM-System die CD-Phase des Cindy4G-Modells ggf. direkt mit einem bereits existierenden Intermediate-VMI beginnen, was die Zyklusdauer zur Bereitstellung der Softwareänderung in der Staging-Umgebung deutlich verringert.

Das Anwendungs-VMI: Ausgehend vom Intermediate-VMI wird schließlich ein Anwendungs-VMI installiert, in dem zusätzlich die aktuelle Version der Software samt aller benötigten Daten enthalten ist. Dieses eignet sich nicht nur zur Bereitstellung eines Staging- bzw. Produktivservers, sondern erlaubt auch eine schnelle Instanziierung von Entwickler-VMs mit der aktuellen Softwareversion, bei denen der langwierige Teil der VM- und Anwendungsinstallation entfällt.

Linux Containers: Der Vollständigkeit halber sei erwähnt, dass Cindy4G grundsätzlich auch die Verwendung der leichtgewichtigeren *Linux Containers (LXC)* an Stelle von VMs unterstützt. In diesem Fall ist allerdings auch die Verwendung anderer Tools, z. B. *Docker* statt *Vagrant*, empfehlenswert. Dementsprechend kämen in diesem Fall statt VM-Images entsprechend Docker-Images zum Einsatz.

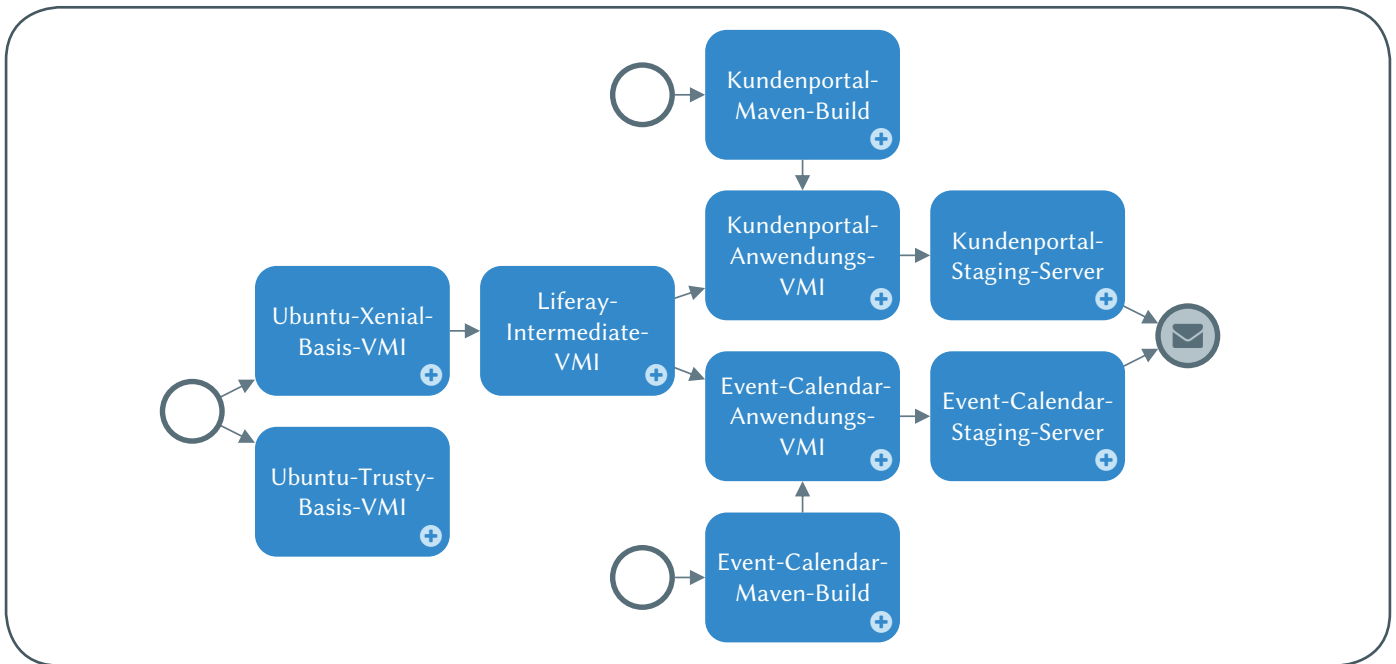
CINDY4G: TECHNISCHE UMSETZUNG FÜR DAS KUNDENPORTAL DER GWDC

Verwendete GWDC-Dienste und Tools

Für die Umsetzung von Cindy4G wurde auf Tools zurückgegriffen, die bereits als GWDC-Dienst angeboten werden oder in anderen Kontexten, z. B. Forschungsprojekten, erfolgreich eingesetzt wurden. So wird als SCM-System und Konfigurationsverwaltung auf *GitLab* zurückgegriffen. Für das Process Management kommt wegen seiner Flexibilität und Übersichtlichkeit vor allem bei CD-Prozessen, die mehrere GitLab-Projekte umfassen, das CD-Tool *Jenkins* [24] zum Einsatz. Als Datenrepositorium hat sich im Java-Umfeld der *NXRM* bewährt, über den die Firma Sonatype derzeit mehr als 2,7 Millionen Softwarebibliotheken, oft auch als *Dependencies* bezeichnet, für Build-Tools wie *Maven* oder *Gradle* [25] weltweit zum Download bereitstellt. Für die VMI-Builds wird das Tool *Packer* [26] in Kombination mit der Virtualisierungsplattform *Kernel-based Virtual Machine (KVM)* [27] eingesetzt. Wie bereits angedeutet kommen die GWDC Cloud Server als IaaS-Plattform sowie *Vagrant* als Tool für die Erstellung von VMs aus den erzeugten VMIs zum Einsatz.

Installation der Systemkomponenten von Cindy4G

Da der *NXRM* und *Jenkins* momentan nicht als regulärer Dienst von Seiten der GWDC angeboten werden, wurden diese auf VMs innerhalb der *GWDC Cloud Server* installiert. Dabei ist insbesondere für den *NXRM* auf einen ausreichend groß dimensionierten Datenspeicher zu achten. Beispielsweise beträgt die Dateigröße eines Anwendungs-VMIs für das Kundenportal knapp



5_Jenkins-Pipelines für das Kundenportal der GWVG und den Göttingen Campus Event Calendar

2,5 GB. Wird der CD-Prozess ggf. noch für verschiedene Git-Branches verwendet, kann der Speicherbedarf des NXRM auch bei einer vergleichsweise kleinen Zahl von Git-Commits pro Tag schnell in den TeraByte-Bereich gelangen. Hier bietet der NXRM allerdings auch diverse Cleanup-Funktionen, welche die Anzahl gespeicherter VMI-Versionen während der Entwicklung auf eine vorgegebene Zahl beschränken und damit einem übermäßigen Verbrauch an Speicherplatz entgegenwirken. Hinsichtlich Rechenleistung und Arbeitsspeicher kommen die VMs mit einer Konfiguration von vier Central Processing Unit (CPU)-Kernen und 4 GB Random-Access Memory (RAM) problemlos zurecht.

Als Build-Server für die Softwarepakete und die VMIs dient derzeit ein physischer Server mit 384 GB RAM und 48 CPU-Kernen, der gleichzeitig auch für die VMI-Builds der *GWVG Cloud Server* verwendet wird. Die üppige Hardwareausstattung bringt für Cindy4G in der Praxis allerdings keine nennenswerten Vorteile; die VMI-Builds für das Kundenportal dauern auf einer Entwickler-Workstation mit lediglich 32 GB RAM und vier CPU-Kernen in etwa gleich lang. Die Verwendung einer VM als Build-Server ist hingegen nur bedingt zu empfehlen, da die durch die verschachtelte Virtualisierung noch verstärkten Leistungseinbußen im I/O-Bereich bei den VMI-Builds besonders deutlich zum Tragen kommen.

Jenkins-Pipelines

Abb. 5 zeigt die ausgehend vom generischen Cindy4G-Modell und dem mehrstufigen VMI-Build tatsächlich implementierten Jenkins-Pipelines zur Umsetzung von Cindy4G für das Kundenportal. Um die Wiederverwendbarkeit des Intermediate-VMIs zu zeigen, wurden zusätzlich Pipelines für den Göttingen Campus Event Calendar realisiert, der ebenfalls einen Liferay-basierten Technologiestack verwendet.

Zur Implementierung von Jenkins-Pipelines stehen eine *Declarative-Pipeline-Syntax* und eine *Scripted-Pipeline-Syntax* zur Verfügung, die sich an den Paradigmen zur deklarativen bzw. imperativen Programmierung orientieren. [28] Beide implizieren durch sogenannte *Stages* eine Gliederung der Pipeline in verschiedene

Bereiche, innerhalb derer ein oder mehrere Tasks, in Jenkins auch als *Steps* bezeichnet, ausgeführt werden. Quelltext 1 zeigt als Beispiel die grundlegende Struktur der Jenkins-Pipelines für die VMI-Builds (ohne die in den Stages enthaltenen Steps).

Bei der Declarative-Pipeline-Syntax handelt es sich um eine *Domain-Specific Language (DSL)*, die sich von ihrer Struktur her auch an Nicht-Programmierer wendet und mit wenig Einarbeitungszeit Ergebnisse liefert. Dafür ist der verfügbare Funktionsumfang eingeschränkt, kann aber durch Plugins um zusätzliche Steps erweitert werden [29]. Außerdem besteht mit dem *Shell-Script-Step (sh)* eine Möglichkeit zur Ausführung von Kommandozeilenbefehlen und Shell- bzw. Interpreter-Skripts. Nicht zuletzt erlaubt der *Script-Step (script)* die Einbettung von Bereichen in der Scripted-Pipeline-Syntax, mittels derer auf einen großen Teil des Funktionsumfangs der Programmiersprache *Groovy* [30] zurückgegriffen werden kann. Damit stehen bei Bedarf auch Programmieretechniken wie Variablenzuweisungen zur Laufzeit, Kontrollstrukturen, Schleifen und Ausnahmebehandlung zur Verfügung.

Bei der Realisierung der Jenkins-Pipelines für Cindy4G wurden verschiedene Best Practices, u. a. des CD-Solution-Providers *CloudBees* [31], berücksichtigt. Hierzu zählen z. B. die Verwendung des offiziellen Pipeline-Plugins oder die Nutzung von SCM-verwalteten *Jenkinsfiles*. Ein besonderes Augenmerk wurde auf die parallele Ausführung von Pipelines und der darin enthaltenen Stages bzw. Steps gelegt. Hierdurch kann Jenkins die anfallende Workload automatisch auf alle verfügbaren Build-Server verteilen, sodass perspektivisch eine unkomplizierte Möglichkeit zur horizontalen Skalierung der Cindy4G-Umgebung besteht.

Abb. 6 zeigt die resultierende grafische Web-GUI-Darstellung der Jenkins-Pipeline aus Quelltext 1 inklusive des parallelen Uploads der erstellten VMIs in die *GWVG Cloud Server* und den NXRM.

Pipeline-Synchronisation mit Locks und Milestones

Die VMI-Erstellung mit Packer innerhalb der *Build-Stage* bedingt u. a. die Ausführung einer VM auf dem Build-Server. Dieser Vorgang ist zeitaufwendig und verursacht eine hohe

Quelltext 1: Jenkins-VMI-Pipeline: grundlegende Struktur und Stages für die VMI-Erstellung

```

1 pipeline {
2   agent any // Auswahl der Build-Server
3   options {} // Globale Optionen für die Pipeline, z. B. Logging oder Timeout
4   parameters {} // Parameter für den jeweiligen Pipeline-Durchlauf
5   environment {} // Umgebungsvariablen für die in den Stages enthaltenen Steps
6   stages { // Pipeline-Aktivitäten
7     stage("Init") {} // Packer-Build vorbereiten
8     stage("Build (Packer)") {} // Packer-Build durchführen
9     stage("Test") {} // VMIs testen
10    stage("Publish") { // VMIs veröffentlichen
11      parallel { // Uploads parallel ausführen
12        stage("Publish (Nexus)") {} // Upload in den NXRM
13        stage("Publish (GWDG-Cloud)") {} // Upload in die GWDG-Cloud
14      }
15    }
16  }
17  post {} // Nachgelagerte Aktionen (E-Mail-Benachrichtigung, Trigger)
18 }

```

Quelltext 1

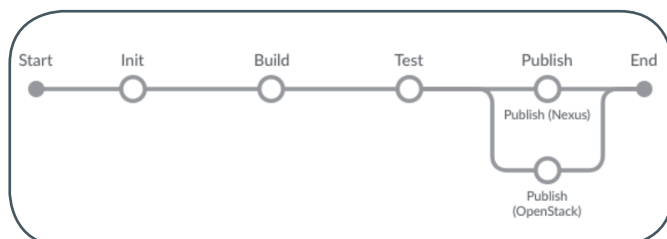
Quelltext 2: Jenkins-VMI-Pipeline: Lock und Milestone für den Packer-Build

```

1 // Lock für Packer-Build reservieren
2 lock(resource: "packer_customer_portal", inversePrecedence: true) {
3   milestone 1 // Ältere Pipeline-Instanzen beenden
4   sh "./packer-build.sh -t -f -V '" // Packer ausführen
5 }

```

Quelltext 2



6_Grafische Darstellung der Jenkins_VMI-Pipeline

Auslastung der Build-Server-Hardware. Da im Zuge von Entwicklungs-Sprints ggf. viele Commits innerhalb kurzer Zeit erfolgen, können die Build-Server durch die parallele Ausführung entsprechend vieler Jenkins-Pipelines mit Packer-Builds an ihre Leistungsgrenzen stoßen.

Um diesem Problem zu begegnen, wurde die Ausführung von Packer-Builds mittels der *Lockable-Resources*- und *Milestone-Step*-Plugins synchronisiert. Als Beispiel dafür wird im Quelltext 2 mit dem *lock*-Step (Zeile 2) zunächst ein exklusiver Zugang zur Ausführung des Packer-Builds angefordert. Sobald einer Pipeline-Instanz Zugang zu diesem Pipeline-Bereich gewährt wurde, werden nachfolgende Instanzen in eine Warteschlange eingereiht.

Diese wird nach dem *Last-In-First-Out*-Prinzip abgearbeitet (*inversePrecedence: true*), sprich nach Freigabe des Locks wird der jeweils aktuellsten Pipeline-Instanz aus der Warteschlange Zugang zum Packer-Build gewährt. Der *milestone*-Step innerhalb des Locks (Zeile 3) wiederum sorgt dafür, dass ältere Pipeline-Instanzen, die sich weiterhin in der Warteschlange befinden und diesen Meilenstein somit noch nicht erreicht haben, automatisch beendet werden.

Durch die Kombination von Locks und Milestones können also die parallele Ausführung gleichartiger sowie die Durchführung veralteter Packer-Builds unterbunden werden. Dieses Muster lässt sich analog auf beliebige zeit- bzw. ressourcenintensive Pipeline-Bereiche anwenden.

ERGEBNISSE UND ZUSAMMENFASSUNG

Ergebnisse

Nach Abschluss der Implementierung aller Pipelines wurde anhand mehrerer Testläufe die Dauer zur Bereitstellung einer Quelltextänderung am Kundenportal in der Staging-Umgebung gemessen. Ausgehend von einem vorhandenen Intermediate-VMI

beträgt diese etwa 70 Minuten; dies ist gleichzeitig der am häufigsten auftretende Fall. Sollten Konfigurationsänderungen am Basis-VMI vorliegen, müssen zusätzlich Basis- und Intermediate-VMI neu erstellt werden. Hierfür werden insgesamt etwas mehr als 2 Stunden Zeit benötigt, also fast doppelt so lange wie bei einem vorhandenen, aktuellen Intermediate-VMI. Da sich die Konfigurationen von Basis- bzw. Intermediate-VMI eher selten ändern, schränkt diese höhere Laufzeit die Eignung von Cindy4G für den Staging-Betrieb aber nicht wesentlich ein.

Die im NXRM befindlichen Anwendungs-VMIs für das Kundenportal lassen sich auch für die Bereitstellung von Entwickler-VMs mit Vagrant und VirtualBox nutzen. Im Vergleich zum Staging-Server werden bei der Instanzierung der VM lediglich einige Konfigurationseinstellungen verändert, beispielsweise für die Netzwerkverbindung. Somit ist eine hohe Übereinstimmung der VMs in Entwicklung, im Staging- sowie im Produktivbetrieb gewährleistet. Im Vergleich zur ursprünglichen Vagrant-Lösung konnte durch die Verwendung des bereits komplett vorinstallierten Anwendungs-VMIs insbesondere die Dauer zur Erstellung einer lokalen Entwickler-VM von bisher ca. 15 Minuten auf knapp 4 Minuten gesenkt werden.

Grundsätzlich sind die Anwendungs-VMIs auch geeignet, ein Update der Produktivumgebung durchzuführen. Hier sind allerdings noch einige Konfigurationsanpassungen vorzunehmen, sodass dieser Schritt weiterhin manuell ausgeführt wird.

Hinsichtlich der an früherer Stelle genannten Anforderungen ist also festzuhalten, dass Quelltextänderungen ausreichend schnell in die Staging-Umgebung Einzug finden, um die angestrebten Ziele wie Transparenz gegenüber Auftraggebern und Projektpartnern zu gewährleisten. Gleichzeitig sorgt die von der Basisinstallation des Server-Betriebssystems über Technologiestack bis hin zur Anwendungsinstallation durchgängige Verwendung von Infrastructure as Code für eine hohe Konsistenz der bereitgestellten VMIs und Server-VMs. Nicht zuletzt profitiert auch die Entwicklung von der Möglichkeit zur schnellen Erstellung von lokalen Entwickler-VMs und der hohen Übereinstimmung mit der Staging- und Produktivumgebung.

Cindy4G wird derzeit parallel zur bisherigen Staging-Umgebung für das Kundenportal und den Event Calendar betrieben und befindet sich somit noch in einer Testphase; ein vollständiger Ersatz der alten Staging-Umgebungen durch Cindy4G ist derzeit in Vorbereitung. Gleichzeitig soll Cindy4G zukünftig sukzessive erweitert werden, z. B. zur Unterstützung von LXC bzw. Docker. Darüber hinaus soll der Einsatz von Cindy4G in Entwicklungs- und Forschungsprojekten jenseits von Kundenportal und Event Calendar weiter evaluiert werden.

Zusammenfassung

In Zeiten immer kürzer werdender Innovationszyklen und zunehmender Komplexität hochgradig vernetzter, cloudbasierter IT-Infrastrukturen ist der Einsatz agiler Vorgehensmodelle und Methoden in der Softwareentwicklung unumgänglich. Dementsprechend wurde von der GWDG mit Cindy4G ein Konzept für eine cloudbasierte Continuous-Delivery-Lösung entwickelt und am Beispiel des Kundenportals implementiert. Abschließend ist festzuhalten, dass Cindy4G die Anforderungen und Erwartungen an die kontinuierliche Bereitstellung einer aktuellen Staging-Umgebung für das Kundenportal der GWDG und den Event Calendar vollumfänglich erfüllt und eine Vielzahl von Erweiterungsmöglichkeiten für den Einsatz in diversen Entwicklungs- und Forschungsprojekten bietet.

LITERATUR

- [AXE] AXELOS. ITIL – IT Service Management [online; besucht am 10. Februar 2018]. Abgerufen unter: <https://www.axelos.com/best-practice-solutions/itil/what-is-itil>
- [DIN15] DIN – DEUTSCHES INSTITUT FÜR NORMUNG E. V. Qualitätsmanagementsysteme – Anforderungen (ISO 9001:2015-11). Beuth Verlag, Berlin, Deutschland, 2015.
- [Fow06] FOWLER, Martin. Continuous Integration [online]. 2006 [besucht am 10. Februar 2018]. Abgerufen unter: <https://martinfowler.com/articles/continuousIntegration.html>
- [FH01] FOWLER, Martin; HIGHSMITH, Jim. The Agile Manifesto [online]. 2001 [besucht am 10. Februar 2018]. Abgerufen unter: <http://www.drdoobs.com/open-source/the-agile-manifesto/184414755>
- [Goo17] GOOGLE CLOUD PLATFORM SOLUTIONS. Automated Image Builds with Jenkins, Packer, and Kubernetes [online]. 2017 [besucht am 10. Februar 2018]. Abgerufen unter: <https://cloud.google.com/solutions/automated-build-images-with-jenkins-kubernetes?hl=en>
- [HF10] HUMBLE, Jez; FARLEY, David. Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation. 1. Aufl., Pearson Education, Boston, MA, USA, 2010. ISBN 978-0-321-60191-9
- [HRN06] HUMBLE, Jez; READ, Chris; NORTH, Dan. The Deployment Production Line. In: Conference on AGILE 2006. IEEE Computer Society, Washington, DC, USA, 2006, S. 113–118. ISBN 978-0-7695-2562-4. Abgerufen unter DOI: [10.1109/AGILE.2006.53](https://doi.org/10.1109/AGILE.2006.53)
- [Le 14] LE CLERC, Cyrille. From Continuous Integration to Continuous Delivery with Jenkins [online]. 2014 [besucht am 10. Februar 2018]. Abgerufen unter: <https://www.javaland.eu/formes/servlet/DocNavi?action=getfile&id=5631548>
- [Lou12] LOUKIDES, Mike. What Is DevOps? [online]. 2012 [besucht am 10. Februar 2018]. Abgerufen unter: <http://radar.oreilly.com/2012/06/what-is-devops.html>
- [Ver14] VERSIONONE. 8th Annual State of Agile Survey [online]. 2014 [besucht am 10. Februar 2018]. Abgerufen unter: <https://www.versionone.com/pdf/2013-state-of-agile-survey.pdf>
- [Zhu14] ZHUO, Zhao. Ensure Continuous Delivery by Deploying Industry Solutions to a Cloud Platform [online]. 2014 [besucht am 10. Februar 2018]. Abgerufen unter: <http://www.ibm.com/developerworks/rational/library/industry-solutions-cloud-platform/index.html>

FUSSNOTEN

- [1] <https://www.gwdg.de/research-education/projects>
- [2] <https://www.gwdg.de>
- [3] GUI: Graphical User Interface
- [4] JEE: Java Platform, Enterprise Edition
- [5] API: Application Programming Interface
- [6] Staging-Umgebung: Entwicklungs- bzw. Vorproduktivumgebung zur Vorbereitung von Versionsupdates für den Produktivbetrieb
- [7] <https://git-scm.com>
- [8] <https://subversion.apache.org>
- [9] <https://www.sonatype.com/nexus-repository-oss>

- [10] <https://puppet.com/products/how-puppet-works>
 [11] <https://www.ansible.com>
 [12] <https://www.chef.io/chef>
 [13] Um Verwechslungen mit der oftmals verwendeten Abkürzung CD für Continuous Delivery zu vermeiden, wird Continuous Deployment in diesem Artikel immer ausgeschrieben.
 [14] <https://www.gwdg.de/server-services/gwdg-cloud-server>
 [15] VM: Virtual Machine
 [16] <https://www.vagrantup.com>
 [17] <https://www.virtualbox.org>
 [18] IBM: International Business Machines Corporation; hier: developerWorks Entwickler-Community (<https://www.ibm.com/developerWorks>)
 [19] CloudBees: Solution-Provider für Enterprise-Jenkins- und DevOps-Lösungen (<https://www.cloudbees.com>)
 [20] <https://www.debian.org/releases/stable/amd64/apb.html>
 [21] <https://events.goettingen-campus.de>
 [22] <https://academiccloud.de>
 [23] <https://vf.u.sofi.gwdg.de>
 [24] <https://jenkins.io>
 [25] <https://gradle.org/maven>
 [26] <https://www.packer.io>
 [27] <https://www.linux-kvm.org>
 [28] <https://jenkins.io/doc/book/pipeline/syntax>
 [29] <https://jenkins.io/doc/pipeline/steps>
 [30] <http://groovy-lang.org>
 [31] <https://www.cloudbees.com/blog/top-10-best-practices-jenkins-pipeline-plugin> ■

Kurz & knapp

Spezialisten für das digitale Zeitalter

Presseinformation Nr. 127 der Georg-August-Universität Göttingen vom 04.06.2018

Das digitale Zeitalter schreitet voran und Fachkräfte mit Datenexpertise sind gefragt. Mit zwei neuen Bachelorstudiengängen bildet die Universität Göttingen ab dem Wintersemester 2018/2019 Spezialistinnen und Spezialisten für „Data Science“ aus. Diese Wissenschaft an der Schnittstelle von Mathematik, Informatik und Statistik beschäftigt sich mit dem Verarbeiten von großen und teilweise unstrukturierten Datenmengen und mit der Analyse und dem Erkenntnisgewinn aus Daten.

Der Bachelorstudiengang „Angewandte Data Science“ vermittelt aufbauend auf den Grundlagen der Informatik und Mathematik vertiefende Kenntnisse der Datenanalyse. Dazu gehört maschinelles Lernen, Statistik, Mustererkennung und die für eine effiziente Analyse benötigten Infrastrukturen. Außerdem erarbeiten sich die Studierenden in einem Anwendungsfach den praktischen Einsatz der erlernten Methoden. Dabei können sie unter Wirtschaft, Biologie, digitalen Geisteswissenschaften, medizinischer Informatik und Züchtungsinformatik wählen.

Der Bachelorstudiengang „Mathematical Data Science“ beschäftigt sich mit den theoretischen Grundlagen von Data Science und richtet sich an Studierende, die ein vertieftes theoretisches Verständnis des Faches suchen. Dazu erwerben sie zunächst fundierte mathematische und statistische Kenntnisse und werden im Verlauf des Studiums in einem Studienschwerpunkt ihrer Wahl an aktuelle Forschungsthemen herangeführt. Absolventinnen und Absolventen sind sowohl auf die Anforderungen eines anschließenden Masterstudiums als auch

auf die Herausforderungen eines Berufseinstiegs hervorragend vorbereitet.

Data Scientists werden derzeit in nahezu allen Fachdisziplinen gesucht. Mögliche Arbeitgeber finden sich unter anderem im Marketing, im IT-Bereich, in Banken und Versicherungen sowie in der Entwicklung und Forschung. Beide Studiengänge sind zulassungsbeschränkt, eine Bewerbung ist noch bis zum 15. Juli möglich. Weiterführende Informationen sind im Internet für Mathematical Data Science unter www.uni-goettingen.de/de/578300.html und für Angewandte Data Science unter www.uni-goettingen.de/de/578300.html zu finden.

Kontaktadressen:

Angewandte Data Science
 Prof. Dr. Jens Grabowski
 Georg-August-Universität Göttingen – Fakultät für Mathematik und Informatik
 Goldschmidtstraße 7, 37077 Göttingen, Telefon (0551) 39-172022
 E-Mail: studiendekan@informatik.uni-goettingen.de
 Internet: www.uni-goettingen.de/de/187936.html

Mathematical Data Science
 Denise Krempasky
 Georg-August-Universität Göttingen – Fakultät für Mathematik und Informatik
 Bunsenstr. 3 bis 5, 37073 Göttingen, Telefon (0551) 39-7762
 E-Mail: studienberatung@math.uni-goettingen.de
 Internet: www.uni-goettingen.de/de/358995.html

ABSCHIED VON OLGA SMIRNOVA

Frau Olga Smirnova hat die GWGD nach knapp sechsjähriger Tätigkeit zum 31. März 2018 verlassen. Sie war seit dem 01.06.2012 als wissenschaftliche Hilfskraft in der Arbeitsgruppe „Nutzerservice und Betriebsdienste“ (AG H) tätig. Frau Smirnova gehörte zum Support-Team und hat Lösungen für GWGD-Nutzer erarbeitet und sie telefonisch sowie remote unterstützt. Sie arbeitete zudem im Bereich SharePoint und erstellte Anleitungen für die Nutzer. Wir danken Frau Smirnova für die langjährige gute Zusammenarbeit und wünschen ihr für ihren beruflichen und privaten Lebensweg weiterhin alles Gute und viel Erfolg.

Helmvoigt



NEUER MITARBEITER SIMON GRAMS

Seit dem 1. Juni 2018 verstärkt Herr Simon Grams die Arbeitsgruppe „Nutzerservice und Betriebsdienste“ (AG H). Herr Grams hat eine Ausbildung zum IT-Systemelektroniker bei bueroboss.de/kassebeer absolviert und anschließend vier Jahre in dem Beruf als Techniker gearbeitet. Sein Aufgabenbereich umfasste die Projektleitung für Auslieferungen, Dispatching und Reparaturarbeiten an Multifunktionsprintern sowie die Betreuung der Kunden. Neben der Ausbildung hat er seine Zertifizierung zum Microsoft Certified IT Professional (MCITP) erworben. Herr Grams wird die Service-Hotline der GWGD unterstützen sowie technische Aufgaben in der Arbeitsgruppe übernehmen. Er ist per E-Mail unter simon.grams@gwdg.de und telefonisch unter 0551 201-1543 erreichbar.

Hast

NEUER MITARBEITER LEON MERTEN LOHSE

Am 1. Mai 2018 hat Herr Leon Merten Lohse seine Tätigkeit als Netzwerkspezialist in der Arbeitsgruppe „IT-Infrastruktur“ (AG I) begonnen. Aufgrund seines erfolgreich absolvierten Bachelor-Studiums im Fach Physik (Röntgenphysik) sowie seiner mehrjährigen Tätigkeit im Bereich der Wohnheimvernetzung ist Herr Lohse für die in der AG I anfallenden Tätigkeiten gut präpariert. Sein Aufgabenschwerpunkt wird im Bereich des Netzwerkmanagements liegen. Herr Lohse ist zunächst bis Ende August 2018 im Netzwerkbereich tätig. Herr Lohse ist per E-Mail unter leon.lohse@gwdg.de und telefonisch unter 0551 201-26837 erreichbar.

ISleiber



NACHRUF MANFRED EYSSELL

Wir trauern um unseren ehemaligen langjährigen Mitarbeiter Herrn Dipl.-Ing. Manfred Eyßell, der am 5. Mai 2018 im Alter von 71 Jahren verstorben ist. Herr Eyßell war vom 1. Juni 1975 bis zu seinem Eintritt in den Ruhestand als wissenschaftlicher Mitarbeiter in unserer Gesellschaft tätig. Während dieser Zeit hat er sich insbesondere um die IT-technische Unterstützung der Institute der Universität Göttingen und der Max-Planck-Gesellschaft gekümmert. Weiterhin hat er entscheidend an der Verwirklichung und der Betreuung des Rechnermuseums der GWGD mitgewirkt. Das Museum war ihm auch in seinem Ruhestand bis zuletzt ein großes Anliegen. Wir werden uns seiner in großer Dankbarkeit erinnern.

Yahyapour, Suren



INFORMATIONEN:

support@gwdg.de
0551 201-1523

Juni bis
Dezember 2018

Kurse

KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
OUTLOOK – E-MAIL UND GROUPWARE	Helmvoigt	14.06.2018 9:15 – 12:00 und 13:00 – 16:00 Uhr	07.06.2018	4
ANGEWANDTE STATISTIK MIT SPSS FÜR NUTZER MIT VORKENNTNISSEN	Cordes	20.06. – 21.06.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	13.06.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	27.06.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	20.06.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	28.06.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	21.06.2018	4
STATISTIK MIT R FÜR TEILNEHMER MIT VOR-KENNTNISSEN – VON DER ANALYSE ZUM BERICHT	Cordes	03.07. – 04.07.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	26.06.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	15.08.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	08.08.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	16.08.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	09.08.2018	4
GRUNDLAGEN DER BILDBEARBEITUNG MIT PHOTOSHOP	Töpfer	21.08. – 22.08.2018 9:30 – 16:00 Uhr	14.08.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	12.09.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	05.09.2018	4

KURS	VORTRAGENDE/R	TERMIN	ANMELDEN BIS	AE
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	13.09.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	06.09.2018	4
INDESIGN – GRUNDLAGEN	Töpfer	18.09. – 19.09.2018 9:30 – 16:00 Uhr	11.09.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	17.10.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	10.10.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	18.10.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	11.10.2018	4
PHOTOSHOP FÜR FORTGESCHRITTENE	Töpfer	23.10. – 24.10.2018 9:30 – 16:00 Uhr	16.10.2018	8
EINFÜHRUNG IN DIE STATISTISCHE DATEN-ANALYSE MIT SPSS	Cordes	13.11. – 14.11.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	06.11.2018	8
ADMINISTRATION VON PCS IM ACTIVE DIRECTORY DER GWDG	Quentin	15.11.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	08.11.2018	4
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	21.11.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	14.11.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	22.11.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	15.11.2018	4
INDESIGN – AUFBAUKURS	Töpfer	27.11. – 28.11.2018 9:30 – 16:00 Uhr	20.11.2018	8
OUTLOOK – E-MAIL UND GROUPWARE	Helmvoigt	06.12.2018 9:15 – 12:00 und 13:00 – 16:00 Uhr	29.11.2018	4
ANGEWANDTE STATISTIK MIT SPSS FÜR NUTZER MIT VORKENNTNISSEN	Cordes	11.12. – 12.12.2018 9:00 – 12:00 und 13:00 – 15:30 Uhr	04.12.2018	8
SHAREPOINT – EINFÜHRUNG FÜR ANWENDER	Buck, Kasper	19.12.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	12.12.2018	4
SHAREPOINT – EINFÜHRUNG FÜR SITECOLLECTION-BESITZER	Buck, Kasper	20.12.2018 9:00 – 12:30 und 13:30 – 15:30 Uhr	13.12.2018	4

Teilnehmerkreis

Das Kursangebot der GWDG richtet sich an alle Mitarbeiterinnen und Mitarbeiter aus den Instituten der Universität Göttingen und der Max-Planck-Gesellschaft sowie aus einigen anderen wissenschaftlichen Einrichtungen.

Anmeldung

Anmeldungen können schriftlich per Brief oder per Fax unter der Nummer 0551 201-2150 an die GWDG, Postfach 2841, 37018 Göttingen oder per E-Mail an die Adresse support@gwdg.de erfolgen. Für die schriftliche Anmeldung steht unter <https://www.gwdg.de/antragsformulare> ein Formular zur Verfügung. Telefonische Anmeldungen können leider nicht angenommen werden.

Kosten bzw. Gebühren

Unsere Kurse werden wie die meisten anderen Leistungen der GWDG in Arbeitseinheiten (AE) vom jeweiligen Institutskontin-

gent abgerechnet. Für die Institute der Universität Göttingen und der Max-Planck-Gesellschaft erfolgt keine Abrechnung in EUR.

Absage

Sie können bis zu acht Tagen vor Kursbeginn per E-Mail an support@gwdg.de oder telefonisch unter 0551 201-1523 absagen. Bei späteren Absagen werden allerdings die für die Kurse berechneten AE vom jeweiligen Institutskontingent abgebucht.

Kursorte

Alle Kurse finden im Kursraum oder Vortragsraum der GWDG statt. Die Wegbeschreibung zur GWDG sowie der Lageplan sind unter <https://www.gwdg.de/lageplan> zu finden.

Kurstermine

Die genauen Kurstermine und -zeiten sowie aktuelle kurzfristige Informationen zu den Kursen, insbesondere zu freien Plätzen, sind unter <https://www.gwdg.de/kursprogramm> zu finden.



Gesellschaft für wissenschaftliche
Datenverarbeitung mbH Göttingen