Erstellung optimierter Executables geschützter Programme im Grid

Christian Boehme, Laurence Finston*, Xin Jin, Michaela Mohr
Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG), Deutschland, lfinsto@gwdg.de

Kurzfassung

Die Installation von Software erfolgt im High-Performance-Computing- (HPC-) Umfeld häufig durch Übersetzung des Quelltextes auf dem Zielsystem, da so ein für dieses optimiertes Executable erzeugt werden kann. Soll der Quelltext eines Programms geschützt werden, muss der Anbieter dem Nutzer bislang generische Executables oder ein eigenes Verfahren zur Verteilung optimierter Versionen bereitstellen. In diesem Paper stellen wir mit dem **Secure-Installer** ein Tool vor, mit dem die nutzergesteuerte Installation von Software im Grid aus dem Quelltext möglich ist, ohne dass der Nutzer Einsicht in den Quelltext bekommt.

1 Einleitung

Im Unterschied zu Desktop-Computern oder auch Standard-Servern zeichnen sich HPC-Ressourcen nach wie vor durch eine große Heterogenität hinsichtlich Hardware und Software aus. Beispiele sind spezielle schnelle Netzwerke zur Parallelverarbeitung oder für die jeweilige Hardware optimierte mathematische Bibliotheken. Mit zunehmender Verbreitung von Graphics Processing Units (GPUs) und ähnlicher Hardware zur Beschleunigung von rechenintensiven Aufgaben – wie zum Beispiel der numerischen Simulation – wird diese Heterogenität wahrscheinlich weiter zunehmen. Daher ist die Bereitstellung optimierter Executables durch Übersetzung des Quelltextes auf der Zielressource nach wie vor von großer Bedeutung. Hinzu kommen mögliche Performance-Gewinne durch die Wahl von zum Zielsystem passenden Compiler-Optionen [1]. Soll dabei aber der Quelltext selbst vor dem Anwender verborgen bleiben, ist der Aufwand für Software-Anbieter oft nicht rentabel.

Im Rahmen des OptiNum-Grid-Projektes [2] wurde mit dem Installer ein Tool entwickelt, das es Nutzern erlaubt, Software auf verteilten Grid-Ressourcen zu installieren. Dazu trägt der Nutzer eine Bezugsquelle (häufig eine URL, andere sind möglich) sowie nach Bedarf ein Installations-Skript (ein Default-Skript ist vorhanden) in eine zentrale Datenbank ein. Die Installation kann dann explizit auf Anforderung des Nutzers oder auch als Präambel eines Grid-Jobs direkt auf dem Frontend der Ressource erfolgen.

Dieses Tool wurde nun so erweitert, dass dem Nutzer Bezugsquelle und Installations-Skript in verschlüsselter Form zur Verfügung gestellt werden, während er keinen Zugriff auf den Quelltext erhält. Dadurch kann es auch im Zusammenhang mit schutzbedürftigem Quelltext eingesetzt werden. Der Vorteil der einfachen Nutzbarkeit für den Anwender bleibt dabei erhalten, während Software-Anbieter von angepassten Executables, einer Zugangskontrolle und der Möglichkeit von Updates durch Modifikation des Quelltext-Repositories profitieren.

2 Der Secure-Installer

2.1 Einleitung

Das ursprüngliche Konzept des Installers sah keine besonderen Sicherheitsvorkehrungen bezüglich der Daten, die in der Software-Datenbank gespeichert werden, oder des Installationsverfahrens vor. Um die Anforderungen bei der Installation von geschützten Quelltexten zu erfüllen wurde nunmehr ein „abgesichertes“ Installationsverfahren als Option hinzugefügt. Bei diesem wird einem Endnutzer durch den Software-Anbieter ein Software-Paket zur Verfügung gestellt, ohne dass der Nutzer erfährt, wie und von woher es heruntergeladen wurde, oder wie die Installation vorgegangen ist. Auch die Quelldateien bleiben ihm verborgen. Der Software-Anbieter bestimmt, was der Nutzer bei erfolgreicher Installation erhält: Kopien von ausführbaren Dateien, eine Software-Bibliothek, Header-Dateien, o.ä.

Um diese Anforderungen zu erfüllen, wurde

1. die Datenbanktabelle **Entries** um die Felder **encrypted**, **download_url_encrypted**, **download_command_encrypted**, **installation_script_encrypted**, **owner** und **authorization** erweitert
2. die Client/Server-Programme **optdbcli** und **optdbsrv** angepasst
3. ein Zusatzprogramm **scrinstl** („Secure Installer“) erstellt, das die Installation (im Normalfall) unter dem Konto des Software-Anbieters auf dem Cluster-Front-End-Rechner (CFE), und daher vor dem Kunden versteckt, vornimmt.

Um sicherzustellen, dass Unbefugte keinen Zugriff auf die geheimzuhaltenden Daten bekommen, werden kryptographische Verfahren mittels des Software-Pakets „The GNU Privacy Guard“ (GPG) eingesetzt. Dies betrifft allein den Software-Anbieter; vom Nutzer werden weder Kenntnisse von GPG noch dessen Vorhandensein auf seinen Computern gefordert. Für eine „abgesicherte“ Installation muss der Software-Anbieter dem Kunden drei Da-

teien zur Verfügung stellen: Die erste dient dem Bezug der Quellen und enthält entweder eine URL oder einen Befehl bzw. eine Befehlsfolge. Die zweite ist ein Shellskript zur Steuerung des eigentlichen Installationsvorgangs und die dritte enthält Informationen zur Autorisierung der Installation. Diese Dateien werden mittels GPG mit dem öffentlichen Schlüssel des Software-Anbieters verschlüsselt und mit dessen privatem Schlüssel zusätzlich signiert.

2.2 Bestandteile

Zwei Server bilden die Basis-Infrastruktur der OptiNum-Grid-Installation der GWDG: **optinum-srv.gwdg.de** dient als Grid-Portal und als Datenbank-Server, **optinum.de** als Web-Service-Portal.

Das Secure Installer-Paket besteht aus folgenden Programmen:

- Dem Client/Server-Paar **optdbcli/optdbsrv**:
  - **optdbcli**: Wird vom Benutzer aufgerufen um Daten in der Software-Datenbank zu speichern und eine Installation auf einem Cluster-Front-End zu veranlassen. Zusätzlich können Informationen über die eigenen Einträge sowie Einträge anderer, die als „public“ markiert wurden, angefordert werden.
  - **optdbsrv**: Läuft auf **optinum-srv.gwdg.de**. Nimmt Befehle und Anfragen von **optdbcli** entgegen und verwaltet die Software-Datenbank.
- **scrinstl**: Führt die „abgesicherte“ Installation aus. Läuft als Daemonprozess auf einem CFE (z.B., rocks-goegrid.gwdg.de), normalerweise (siehe unten) unter dem Benutzerkonto eines Software-Anbieters.
- **optwbsrv**: Eine Webanwendung, die auf **optinum.de** läuft und **optdbcli** auf dieser Maschine indirekt über **optwbsub** (siehe Abschnitt 2.3.1.4) aufruft. Ermöglicht denjenigen die Nutzung von **optdbcli**, die dieses Programm nicht auf ihren Rechnern installieren können oder wollen. Zur Zeit betrifft dies insbesondere Nutzer von Windows-Systemen (oder anderer Nicht-UNIX-ähnlicher Systeme), jedoch sollte eine Portierung des Secure Installer-Pakets insgesamt auf Windows oder andere Systeme mit vertretbarem Aufwand möglich sein.
- **optwbsub**: Läuft als Daemonprozess auf **optinum.de**. Wird von **optwbsrv** aufgerufen und ruft seinerseits **optdbcli** auf.

Zusätzlich stehen eine grafische Benutzeroberfläche (GUI) für Datenbankeinträge sowie Tools zur Durchführung einer Installation auf der Kommandozeile bzw. integriert in den Job-Scheduler **GridWay** zur Verfügung. Sie werden in den Abschnitten 2.7 und 2.8 beschrieben.

2.3 Software-Datenbank

Die MySQL-Datenbank **dbsrvcli** befindet sich auf **optinum-srv.gwdg.de** und wird vom Server-Programm **optdbsrv** verwaltet. Sie enthält Tabellen und Daten für die Benutzerverwaltung und Informationen über Software-Pakete für deren Installation (siehe Tabelle 1).

Tables_in_dbsrvcli
Certificates
Delegates
Entries
Prerequisites
Public_Keys
Users
Users_Certificates

Tabelle 1: Tabellen der Datenbank für Software-Pakete

Informationen für die Installation eines Software-Pakets werden in der Tabelle **Entries** gespeichert (siehe Tabelle 2).

Field	Type
user_id	int(11)
entry_id	int(11)
package_name	varchar(256)
package_version	varchar(256)
version_ctr	int(11)
download_url	varchar(256)
download_command	varchar(256)
installation_script	mediumtext
maintainer_name	varchar(256)
maintainer_email_address	varchar(256)
private	tinyint(1)
checked_by_admin	tinyint(1)
created	datetime
last_modified	datetime
timezone	char(3)
encrypted	tinyint(1)
download_url_encrypted	blob
download_command_encrypted	blob
installation_script_encrypted	blob
owner	varchar(256)
authorization	blob

Tabelle 2: Felder der Tabelle Entries der Datenbank für Software-Pakete

Die Felder **download_command**, **download_url** und **installation_script** werden nur bei Software-Paketen verwendet, die „normal“, d.h. ohne besondere Sicherheitsmaßnahmen, installiert werden. Für Software-Pakete dagegen, die „abgesichert“ installiert werden, werden stattdessen die Felder **download_command_encrypted**, **download_url_encrypted** sowie **installation_script_encrypted** verwendet. In beiden Fällen enthalten die jeweils nicht benutzten Felder den Wert NULL.

2.3.1 Eintragungen in die Datenbank

Für Eintragungen in die Software-Datenbank ruft der Benutzer das Client-Programm **optdbcli** von einem beliebigen Rechner aus auf, wobei entweder Kommandozeilenoptionen oder die GUI-Komponente verwendet werden können. Um sich dem Server-Programm **optdbsrv** auszuweisen, benötigt er ein X.509-Zertifikat, dessen Herausgeber („Issuer“) DFN-Verein PCA Grid - G01 sei:
organization: DFN-Verein
organizationalUnitName: DFN-PKI
commonName: DFN-Verein PCA Grid - G01
countryName: DE
sowie den Schlüssel, der zum Benutzer-Zertifikat gehört. Das CA-Zertifikat des DFN-Vereins muss ebenfalls auf dem Rechner des Benutzers vorhanden sein, da sich das Server-Programm **optdbsrv** zusätzlich gegenüber dem Client-Programm **optdbcli** mit einem von dieser CA ausgegebenen Server-Zertifikat ausweist.

2.3.1.1 Normale Installation

Wenn ein Software-Paket später „normal“, d.h. ohne besondere Sicherheitsmaßnahmen, installiert werden soll, muss der Benutzer angeben, wie und von welchem Ort die Quellen der Software bezogen werden. Dies kann ein URL (**download_url**) oder ein Befehl bzw. eine Befehlsfolge (**download_command**) sein. Es können auch beide angegeben werden, aber bei der Installation hat dann der **download_command** Vorrang. In der Eingabe zu **optdbcli** wird **download_url** und/oder **download_command** als Zeichenkette übergeben und an **optdbsrv** übertragen; da sie normalerweise nicht sehr umfangreich sein sollten, werden sie nicht in temporären Dateien zwischengespeichert.

Bei einer „normalen“ Installation kann der Benutzer zusätzlich ein Installations-Skript angeben, dessen Inhalt dann im Feld **installation_script** gespeichert wird. Wenn er keines angibt, wird bei der Installation ein Default-Skript verwendet, das für einfache, typische Fälle ausreicht und auch als Vorlage für Anpassungen verwendet werden kann. Durch die Möglichkeit der Verwendung eines eigenen Installations-Skriptes lassen sich auch komplexere Installationen durchführen. Alle Fehlermeldungen und sonstigen Ausgaben werden an den Anwender zurückgegeben. Im Gegensatz zu **download_url** oder **download_command**, muss man für **installation_script** einen Dateinamen angeben.

2.3.1.2 Abgesicherte Installation

Anders verhält es sich bei den Eintragungen für Pakete, die später „abgesichert“ installiert werden. Statt **download_command**, **download_url** und **installation_script** werden die Felder **download_command_encrypted**, **download_url_encrypted** bzw. **installation_script_encrypted**, sowie zusätzlich das Feld **authorization**, verwendet. Für all diese Felder werden nur Dateinamen übergeben. Die Dateien werden für den Kunden vom Software-Anbieter erstellt, verschlüsselt, signiert und ihm übergeben, z.B. per E-Mail.

Der Kunde erstellt dann genauso wie für ein „normal“ zu installierendes Software-Paket, bis auf die Verwendung der erwähnten anderen Felder, mit **optdbcli** einen Datenbankeintrag.

2.3.1.3 Prerequisites

Die Installation eines Software-Pakets kann vom Vorhandensein eines oder mehrerer anderer abhängen. Für diesen Fall ist die Eintragung von Abhängigkeiten in die Software-Datenbank vorgesehen, indem man „prerequisites“, also „Voraussetzungen“ angibt.

Zum Beispiel, die folgende Eingabe besagt, dass **bison 2.4.3** von **m4 1.4.15** abhängt:

```
START_ENTRY
PACKAGE_NAME m4
PACKAGE_VERSION 1.4.15
DOWNLOAD_URL
ftp://ftp.gwdg.de/pub/gnu/ftp/gnu/m4/m4-latest.tar.gz
END_ENTRY

START_ENTRY
PACKAGE_NAME bison
PACKAGE_VERSION 2.4.3
DOWNLOAD_URL http://ftp.gnu.org/gnu/bison/bison-2.4.3.tar.gz
PREREQUISITE m4 1.4.15
PUBLIC
END_ENTRY
```

Mit dem ADD PREREQUISITE-Befehl kann außerdem einem Eintrag nachträglich eine „prerequisite“ hinzugefügt werden. Dieser Befehl kann beliebig oft in der Eingabe verwendet werden und es gibt keine Einschränkung bezüglich der Anzahl der „prerequisites“ für ein gegebenes Paket.

Wenn ein Paket als Voraussetzung angegeben wird, muss ein Eintrag für dieses bereits existieren. Es ist also nicht möglich, ein der Datenbank „unbekanntes“ Paket als Voraussetzung anzugeben.

Wenn später **bison 2.4.3** installiert werden soll, wird ggf. **m4 1.4.15** vorher installiert. D.h., wenn **optdbcli** mit der Option **--install** aufgerufen wird, werden die Pakete nur dann installiert, wenn sie nicht schon auf dem Rechner installiert (und gefunden) worden sind. Wenn es aber mit der Option **--reinstall** aufgerufen wird, werden sie auf jeden Fall neu installiert. Diese Optionen gelten aber für alle Pakete, die bei einem einzelnen Aufruf von **optdbcli** installiert werden (oder nicht). Eine feinere Unterscheidung ist derzeit nicht vorgesehen.

Hinsichtlich der Voraussetzungen spielt es keine Rolle, ob die Software-Pakete „normal“ oder „abgesichert“ installiert werden. Wenn eine „Kette“ von Paketen installiert wird, wird jeweils die erforderliche Art von Installation vorgenommen.

2.3.1.4 Aufruf von **optdbcli** mittels der Webanwendung **optwbsrv**

Für Benutzer, die **optdbcli** auf ihren Rechnern nicht installieren können oder wollen, besteht außerdem die Möglichkeit, es indirekt über eine auf der Maschine **opti-**

num.de installierte Webanwendung, **optwbsrv**, aufzurufen. Dazu verwendet man das verbreitete Dienstprogramm **curl** [3] um Daten mit der Webanwendung auszutauschen. Diese ruft ihrerseits (ebenfalls indirekt über das Programm **optwbsub**) **optdbcli** auf **optinum.de** im Auftrag des Benutzers auf und schickt die Ausgabe von **optdbcli** an diesen zurück (siehe Bild 1). Der notwendige **curl**-Aufruf kann auch mittels der GUI-Komponente generiert werden.

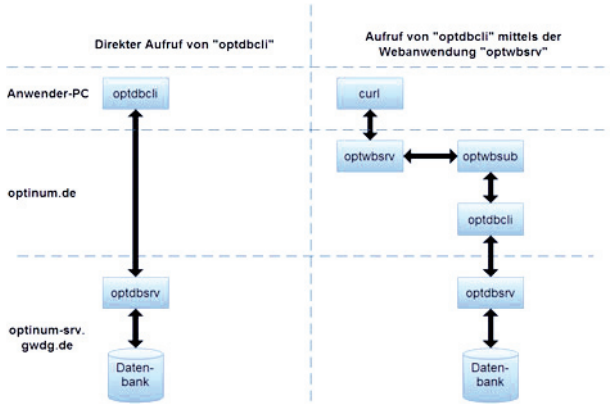

Bild 1: Eintragungen in die Datenbank für Software-Pakete: Die Eintragung kann entweder über einen direkten Aufruf von optdbcli von einem Anwender-PC aus erfolgen oder mittels der Webanwendung optwbsrv.

2.4 Installation

Um ein oder mehrere Software-Pakete auf einem CFE-Rechner zu installieren, ruft ein Benutzer, direkt oder mittels des **InstallStarter**-Programms (siehe Abschnitt 2.7), das Client-Programm **optdbcli** auf.

Um auf den CFE-Rechner zu gelangen, benötigt der Benutzer in der Regel ein Proxy-Zertifikat, das z.B. durch das Werkzeug **grid-proxy-init** erzeugt wird.

Das Proxy-Zertifikat wird per Default von Globus-Toolkit nicht auf den CFE-Rechner übertragen. Um einem Benutzer Eintragungen in der Software-Datenbank zwecks Installation zuzuordnen, wurde früher **optdbcli** dessen „Distinguished Name“ als Argument übergeben. Dieser konnte zwar gefälscht werden, aber das Risiko wurde als gering erachtet, da es nicht vorgesehen war, dass sich geheime Daten in der Software-Datenbank befinden würden. Insbesondere war es nicht vorgesehen, dass Benutzern Daten in den eigenen Eintragungen geheimgehalten werden sollten.

Durch Fälschung eines „Distinguished Names“ hätte man allerdings auch Zugriff auf Einträge erhalten, die als „privat“ markiert worden waren, was allein schon einen besseren Zugriffsschutz wünschenswert erscheinen ließ.

Dieses Problem wurde nunmehr dadurch gelöst, dass das InstallStarter-Programm das Proxy-Zertifikat auf den CFE-Rechner überträgt und somit **optdbcli** und **scrinstl** eine sichere Authentifizierung und Autorisierung des Nutzers ermöglicht.

Wenn dieser sofort oder zu einem späteren Zeitpunkt die Installation auf einer Grid-Ressource veranlasst, erkennt das Client-Programm **optdbcli** anhand der benutzten Felder des Datenbankeintrags, ob es sich um ein Software-Paket handelt, das eine „abgesicherte“ Installation erfordert oder nicht.

Bei einer „normalen“ Installation, steuert **optdbcli** (dessen Prozess unter dem Konto des Benutzers läuft) den gesamten Installationsvorgang, die heruntergeladenen Quelldateien sowie die Ergebnisse der Installation werden in Verzeichnissen unter dem Home-Verzeichnis des Benutzers gespeichert (**~/installer_work/** bzw., per Default, **~/Installer/**). Auch die Ausgabe des Installations-Skriptes, das vom Benutzer gestellt werden kann, ist für diesen sichtbar.

Anders verhält es sich bei einer „abgesicherten“ Installation: **optdbcli** erkennt anhand der Informationen über das Paket, die von **optdbsrv** übersandt werden, dass es sich um einen „verschlüsselten“ („encrypted“) Eintrag handelt. **optdbcli** nimmt dann über eine (UNIX-Domain-) Socket-Datei Kontakt zum Programm **scrinstl**, das als Daemonprozess (im Normalfall, siehe den folgenden Absatz) unter dem Konto des Software-Anbieters läuft, und übersendet diesem die nötigen Informationen, damit es die Installation ausführen kann.

Es ist nicht unbedingt nötig, dass eine Instanz von **scrinstl** unter dem Konto eines jeden Software-Anbieters läuft. Der Name der Socket-Datei, über die der Kontakt zum **scrinstl**-Prozess erfolgt, lautet **scrinstl.<Benutzername des Software-Anbieters>**, z.B., **scrinstl.dgon0015** auf **rocks-goegrid.gwdg.de**. Wenn aber ein anderer Software-Anbieter, z.B., mit Benutzer-ID **dgon0020**, keine eigene Instanz von **scrinstl** laufen lassen will, kann er einen symbolischen Link namens **scrinstl.dgon0020** erstellen, der auf die Socket-Datei **scrinstl.dgon0015** verweist.

Wenn ein Kunde des Software-Anbieters mit Benutzerkonto **dgon0020** ein von diesem gestelltes Software-Paket installieren will, nimmt **optdbcli** Kontakt zu **scrinstl** über den symbolischen Link **scrinstl.dgon0020** auf. Für **optdbcli** spielt es keine Rolle, dass die entsprechende Instanz von **scrinstl** unter dem Benutzerkonto eines anderen Software-Anbieters läuft. Voraussetzung dafür, dass dies funktioniert, ist jedoch, dass der Benutzer **dgon0015** über den privaten Schlüssel des Software-Anbieters mit Benutzername **dgon0020** verfügt.

Dafür, dass **scrinstl** als Daemonprozess immer verfügbar bleibt, sorgt das Shellskript **restart_scrinstl.sh**, das per cronjob in regelmäßigen Abständen (etwa jede Viertelstunde) aufgerufen wird. Dieses prüft, ob die Socket-Datei noch vorhanden ist und der Prozess noch läuft, und startet diesen neu falls das erforderlich ist.

2.5 Ablauf der abgesicherten Installation

Folgende Informationen werden von **optdbcli** an **scrinstl** über die Socket-Datei übersandt:

1. Der Benutzername des Benutzers, der **optdbcli** aufgerufen hat, d.h. des „Kunden“.
2. Paketname (erforderlich).

3. Paketversion (oder „none“, falls keine).
4. „Owner“, d.h. der „Distinguished Name“ des Software-Anbieters.
5. Die Kennzahl des GPG-Schlüsselpaars des Software-Anbieters. (Diese wird in der Software-Datenbank in der Tabelle **Public_Keys** gespeichert.)
6. Verzeichnis des Kunden, wo generierte Dateien u.ä. installiert werden sollen.
7. **download_command** oder 0 (Null).
8. **download_url** oder 0 (Null).
9. **installation_script** oder 0 (Null). Zur Zeit muss eines vorhandenen sein (siehe unten).
10. „Distinguished Name“ des Benutzers (Kunden).
11. Pfad der Datei, die die Autorisierung enthält.

Die Informationen zum Herunterladen des Paketes, **download_url** oder **download_command**, das Installations-Skript sowie die Autorisierung werden nur mit dem Schlüsselpaar des Software-Anbieters verschlüsselt und signiert. Es muss deshalb sichergestellt werden, dass der Kunde, der Kontakt zu **scrinstl** über **optdbcli** aufnimmt, berechtigt ist, die Installation zu veranlassen und Zugriff auf Programme, Dateien o.ä. zu bekommen.

Die Benutzernamen von Kunden und Software-Anbietern können sich von Ressource zu Ressource unterscheiden und sind dort auch nicht unbedingt unveränderlich. Zur Identifizierung dient deshalb jeweils der „Distinguished Name“. Die Zuordnung zum jeweiligen Benutzerkonto auf dem CFE-Rechner erfolgt über das **grid-mapfile**, das jedoch Globus-Toolkit gehört; falls der CFE-Rechner nicht gleichzeitig ein Globus-Front-End-Rechner ist, muss folglich das **grid-mapfile** in regelmäßigen Abständen (zweckmäßigerweise per cronjob) auf jenen kopiert werden.

Im Normalfall muss sich der Kunde mit einem Proxy-Zertifikat ausweisen, um sich überhaupt auf den CFE-Rechner einzuloggen. Um aber ganz sicherzugehen, dass es sich tatsächlich um den richtigen Benutzer handelt, prüft sowohl **optdbcli** als auch **scrinstl** die Gültigkeit des Proxy-Zertifikats, bevor die Download-Informationen, das Installations-Skript und die Autorisierung geprüft werden.

Die Autorisierung enthält folgende Informationen auf genau vier Zeilen:

1. den „Distinguished Name“ des Benutzers, wie er im Grid-Zertifikat und in dem **grid-mapfile** erscheint
2. den Namen und ggf. die Versionsnummer der Software,
3. die sha1-Prüfsumme der verschlüsselten Daten mit den Informationen zum Quellenbezug und
4. die sha1-Prüfsumme des verschlüsselten Installations-Skriptes.

Der Download-Befehl oder -URL, das Installations-Skript sowie die Autorisierung werden entschlüsselt und die Signaturen geprüft. Neue Prüfsummen werden ermittelt und verglichen mit denen, die sich in der Autorisierung befinden.

Der „Distinguished Name“ des Kunden ist das einzige Element, das eine Zuordnung der übergebenen Daten zu einem bestimmten Benutzer auf dem CFE-Rechner erlaubt, es sei denn, der Download-Befehl, -URL, und/oder das Installations-Skript würden den „Distinguished Name“ oder irgendeinen anderen Hinweis enthalten.

Für den Kunden unsichtbar entschlüsselt **scrinstl** die Installationsinformationen und führt die Übersetzung des Quelltextes durch. Anschließend werden die Dateien, die der Software-Anbieter für den Kunden bestimmt hat, an geeignete Stellen kopiert, symbolische Links erstellt (sofern **optdbcli** nicht mit der Option **--no-links** aufgerufen wurde) und alle anderen vom Software-Anbieter vorgesehenen Aktionen ausgeführt.

Eine Installation könnte allein aus dem Herunterladen eines Pakets bestehen, weshalb für ein Installations-Skript keine prinzipielle Notwendigkeit besteht. Zum gegenwärtigen Zeitpunkt aber muss eines existieren, auch wenn es keine Aktionen ausführt, da die Autorisierung eine Prüfsumme dafür enthalten muss.

2.6 Sicherheit

Die Sicherheit des Secure Installer-Pakets hängt von der Sicherheit des CFE-Rechners ab. D.h., die Zugriffsrechte der Verzeichnisse und Dateien unter dem Benutzerkonto des Software-Anbieters müssen korrekt gesetzt sein. Eine Verschlüsselung des Filesystems, der Ein- und Ausgabe, des Swap-Bereiches oder dergleichen liegt nicht im Aufgabenbereich des Secure Installer-Pakets. Folglich muss man davon ausgehen, dass **root** unbegrenzten Zugriff auf Daten des Software-Anbieters auf dem CFE-Rechner hat. Folglich muss ein Software-Anbieter, der das Secure Installer-Paket verwenden will, von der Vertrauenswürdigkeit des Administrators oder der Administratoren des CFE-Rechners, d.h. der Integrität der Grid-Infrastruktur insgesamt, überzeugt sein.

Der Software-Anbieter ist nicht an der Installation persönlich beteiligt und muss nicht zur Installationszeit am CFE-Rechner angemeldet sein. Deshalb kann sein privater GPG-Schlüssel nicht mit einer Passphrase geschützt sein, da er sonst diese bei der Installation eingeben müsste, wiewohl es im Allgemeinen unbedingt empfehlenswert ist, Schlüssel durch eine Passphrase zu schützen. Deshalb sollte man Schlüssel, die vom Secure Installer-Paket verwendet werden, nicht für andere Zwecke einsetzen, sehr darauf bedacht sein, dass sie nicht kompromittiert werden und sie schnell austauschen, sollte dies doch geschehen.

2.7 Integration in die Grid-Infrastruktur

Der eigentliche Installationsprozess erfolgt immer auf dem Frontend einer HPC-Ressource, da hier – üblicherweise – die Umgebung für die Übersetzung von Software für die speziellen Eigenschaften der jeweiligen HPC-Ressource vorhanden ist. Es gibt zwei Möglichkeiten, die Installation der Software zu veranlassen. Mit Hilfe des **InstallStarterClient**, einem Java-Kommandozeilen-Client, lässt sich ein Software-Paket unter Angabe des Globus-

Servers eines Grid-Ressourcenanbieters auf der entsprechenden Ressource installieren. Dieser Client ist auf dem Portal des OptiNum-Grid-Projektes, **optinum-srv.gwdg.-de**, verfügbar. Wenn eine lokale Globus-Installation vorhanden ist, kann er auch heruntergeladen und von dem lokalen Rechner aus aufgerufen werden.

Die zweite Möglichkeit, die Installation der Software zu veranlassen, ist, diese nur dann wenn ein Job auf einer Ressource ausgeführt wird, zu installieren. Diese Möglichkeit besteht wenn ein Job über die GridWay-Installation des OptiNum-Grid-Projektes oder eine andere entsprechend angepasste GridWay-Installation submittiert wird. Die Submittierung erfolgt über ein Shellskript, das eine Datei mit Beschreibung der zu installierenden Software, sowie ein GridWay-Job-Template übergeben bekommt, und dann den GridWay-Job submittiert. Vor Beginn des eigentlichen Jobs wird die zu installierende Software auf der von GridWay für den Job ausgewählten Ressource installiert. Sollte die Software-Installation oder der eigentliche Job fehlschlagen werden sowohl Software-Installation als auch Job auf einer alternativen Ressource gestartet.

Die Durchführung der Installation ist an ein D-Grid-Nutzerzertifikat gebunden. Sowohl für den Aufruf des **InstallStarterClient**, als auch für die Submittierung eines Jobs über GridWay muss sich der Nutzer durch ein Proxy-Zertifikat authentifizieren. Bei beiden Installationsvarianten wird letzten Endes ein Globus-Fork-Job auf dem Globus-Server des Grid-Ressourcenanbieters gestartet. Dieser Fork-Job sorgt dann wiederum für den Aufruf des eigentlichen Installationsprogramms (**optdbcli**) auf dem Frontend der HPC-Ressource. Das Proxy-Zertifikat des Nutzers wird dabei auf das Frontend der HPC-Ressource übertragen und dort dafür genutzt, sich bei dem Installationsprogramm **optdbcli** zu authentifizieren.

2.8 Grafische Nutzeroberfläche

Bild 2. Die GUI identifiziert den Nutzer anhand seiner Grid-Credentials.

Zur Erleichterung der Software-Installation für den Nutzer wird eine grafische Benutzeroberfläche (GUI) zur Verfügung gestellt (siehe Bild 2). Es handelt sich hierbei um eine einfach zu bedienende, plattformunabhängige Java-Anwendung, die auf einem Desktop-PC oder Notebook des Nutzers installiert werden kann. Mittels der GUI werden die zur Software-Installation benötigten Informationen in die zentrale Datenbank des Installers eingetragen. Der Zugriff auf die Datenbank erfolgt nach Verifizierung der Benutzer-Identität durch ein Grid-Zertifikat. Mittels der GUI können auch die in der Datenbank eingetragenen Informationen jederzeit abgefragt und aktualisiert werden.

3 Ausblick

Der Secure-Installer wird zur Zeit gemeinsam mit der ERAS GmbH für eines ihrer Produkte getestet. Der nächste Schritt in der Entwicklung des Secure-Installers wird seine Integration in den ebenfalls aus dem OptiNum-Grid-Projekt stammenden **GridWorker** [4] sein. Hierdurch wird eine deutlich bessere Integration in den Arbeitsablauf typischer Anwender erreicht. Eine Verbesserung der Sicherheit lässt sich durch Verschlüsselung des Quelltextes auf dem Frontend der Zielressource erzielen. Dann ist dieser auch vor Zugriff durch den Systemadministrator bzw. bei einer Privilegieneskalation geschützt. Es soll untersucht werden, ob hierfür der im OptiNum-Grid entwickelte **SecureStorage**-Dienst verwendet werden kann. Es sind auch weitere Möglichkeiten bei der Bedienung des Secure-Installers denkbar. Unter [5] existiert für Testzwecke ein einfaches Webformular, welches bei Bedarf ausgebaut werden könnte.

[1] Katherine E. Stewart , Steven W. White, The Effects of Compiler Options on Application Performance, Proceedings of the 1994 IEEE International Conference on Computer Design: VLSI in Computer & Processors, p.340-343, October 10-12, 1994.

[2] OptiNum-Grid: http://www.optinum.de. BMBF Förderkennzeichen 01IG09011.

[3] http://curl.haxx.se/

[4] Schneider, A.: Variantensimulation mit GridWorker. ASIM-Workshop "Simulation technischer Systeme – Grundlagen und Methoden in Modellbildung und Simulation", Krefeld, 24.-25. Februar 2011.

[5] https://www.optinum.de/optintfc.html