

Universität Siegen  
Compilerbau und Softwareanalyse

## Seminar “Softwarearchitekturen” KDE Software Compilation

Viktor Dück  
vdueck@informatik.uni-siegen.de

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>3</b>
<b>2</b>	<b>Softwarearchitektur</b>	<b>3</b>
<b>3</b>	<b>Entwicklung von KDE</b>	<b>4</b>
<b>4</b>	<b>Qt</b>	<b>5</b>
4.1	Qt Module . . . . .	6
4.2	KDE Free Qt Foundation . . . . .	6
4.3	Zukunft . . . . .	7
<b>5</b>	<b>Linux und KDE</b>	<b>7</b>
<b>6</b>	<b>KDE SC</b>	<b>8</b>
6.1	KDE Workspaces . . . . .	8
6.1.1	Plasma Desktop . . . . .	9
6.1.2	Plasma Netbook . . . . .	9
6.1.3	Plasma Mobile . . . . .	9
6.2	KDE Platform . . . . .	9
6.2.1	Plasma . . . . .	10
6.2.2	Oxygen . . . . .	10
6.2.3	Solid . . . . .	11
6.2.4	Phonon . . . . .	11
6.2.5	Nepomuk . . . . .	11
6.2.6	Akonadi . . . . .	12
6.2.7	Decibel . . . . .	12
6.3	KDE Applications . . . . .	13
<b>7</b>	<b>Aussicht</b>	<b>13</b>

# 1 Einleitung

Im Rahmen des Seminars "Softwarearchitekturen" des Fachbereiches „Compilerbau und Softwareanalyse“ beschäftigt sich diese Ausarbeitung mit der "KDE Software Compilation". Um dem Rahmen gerecht zu werden, wird zunächst das Thema der Softwarearchitektur grob aufgerollt und die Entwicklungsgeschichte von KDE angerissen. Das für KDE grundlegende Qt wird vorgestellt und anschließend wird der Aufbau, der sich mit der Umstrukturierung zur Version 4 ereignet hat, erläutert.

## 2 Softwarearchitektur

Laut Gernot Starke und Peter Hruschka beschreibt der Terminus „Softwarearchitektur“ „die Strukturen des Systems, dessen Bausteine, Schnittstellen und deren Zusammenspiel.“ [29] Ähnlich definiert die Gesellschaft für Informatik Softwarearchitektur :“ Die Software-Architektur ist die grundlegende Organisation eines Systems, dargestellt durch dessen Komponenten, deren Beziehungen zueinander und zur Umgebung sowie die Prinzipien, die den Entwurf und die Evolution des Systems bestimmen. „[8] Folglich sagt die Softwarearchitektur nichts über den detaillierten Entwurf oder gar den Implementationsdetails aus, sondern beschreibt die Zusammenhänge des Systems und gilt als „Kommunikationsmedium“ [8] für die am Projekt beteiligten Personen, wie Projektleiter, Kunden, Entwickler, Tester usw.

Technische Rahmenbedingungen, die die Entwicklung der Architektur beeinflussen sind: [8]

- Standards für Entwicklungsvorgehen (Prüfungen, Richtlinien, Art der Dokumentation)
- Vorgaben für die Entwicklungsumgebung (Programmiersprache, Werkzeuge etc.)
- Vorgaben für die Zielplattform (Prozessor, Speicher, Netzanbindung, Betriebssystem)
- Vorgaben für die zu verwendende Komponenten (Datenbank, Webserver, Klassenbibliothek, Benutzungsschnittstelle)
- Technische Standards (wie Referenzarchitekturen, Datenformate, Kommunikationsprotokolle, Kompatibilität)

Insbesondere ist eine gute Softwarearchitektur bei mittlerem bis großen bzw. langläufigen Projekten von Vorteil, da die Weiterentwicklung an einem durchdachtem System leichter fällt, als an einem, das keiner Struktur folgt. Parallel zur Baukunst gibt es auch in der Softwarearchitektur verschiedenste Architekturen. Bekannt sind die Schichtenarchitektur, das MVC Schemata<sup>1</sup> oder auch im Allgemeinen die SOA, die Serviceorientierte Architektur.

### 3 Entwicklung von KDE

1996 fällt Matthias Ettrich bei der Konfiguration des Rechners seiner Freundin auf, dass Linux auf dem Desktop nicht einheitlich ist. Es gibt verschiedenste Arten von Menüs, Fensterarten, Scrollbars und ebenso ist das Verhalten auf Mausbewegungen und Tastatureingaben unterschiedlich. Beispielsweise ist bei manchen Fenstern das Scrollen nur mit der Mitteltaste möglich und bei anderen fehlt die Reaktion auf die Tasten END und POS1. Obwohl bereits einige Arbeitsumgebungen existieren wie zu Beispiel das proprietäre CDE (Common Desktop Environment) [7] und „Looking Glass“, sowie Window Manager, wie Motif, die jedoch keine GUI (Graphical User Interface) anbieten, gibt es nichts Einheitliches, was dem Nutzer das Benutzen des Computers vereinfacht. So verfasst Matthias Ettrich einen Aufruf für eine neue Arbeitsumgebung. Kool Desktop Environment, kurz KDE, soll es dem normalen Nutzer ermöglichen seinen täglichen Aufgaben und Arbeiten am Computer leichter nachzukommen. Zielgruppe ist ausschließlich der normale Benutzer, „who wants to browse the web with Linux, write some letters and play some nice games.“[4] Zumal bezüglich der Administration es sinnvoller ist, weiterhin mit der UNIX-CLI zu arbeiten, weil das Abbilden der unzähligen Tools mit einer GUI, zum einen keinen Mehrwert bringt und zum anderen ein ungeheures Unterfangen ist.

Ziel ist es also eine Arbeitsumgebung zu schaffen, in der nicht die verschiedensten Bibliotheken mit den eigenen Widgets in den Arbeitsspeicher geladen werden müssen. Man vermeidet damit, dass als Beispiel XV, ein Tool zur Anzeige und Bearbeitung von Bildern und LyX, ein LaTeX-Tool, alle ihre Abhängigkeiten laden und am besten noch von einem Window Manager verwaltet werden, der wiederum weitere Bibliotheken lädt. Aufsetzen soll KDE auf Qt, ein von „Trolltech“ und später von NOKIA übernommene C++ Klassenbibliothek, die das Programmieren von Oberflächen für den X-Server vereinfacht. Im Umfang ist ein Panel nach dem Vorbild von CDE angedacht, aus dem sich Anwendungen starten und weitere Informationen, wie Uhrzeit und Datum ablesen lassen. Ferner sollen weitere Programme entwickelt werden:

---

<sup>1</sup> Model View Controller

1. Dateimanager
2. Mailclient
3. Texteditor
4. Terminal
5. Bildbetrachter
6. Window Manager
7. SystemTools
8. Spiele

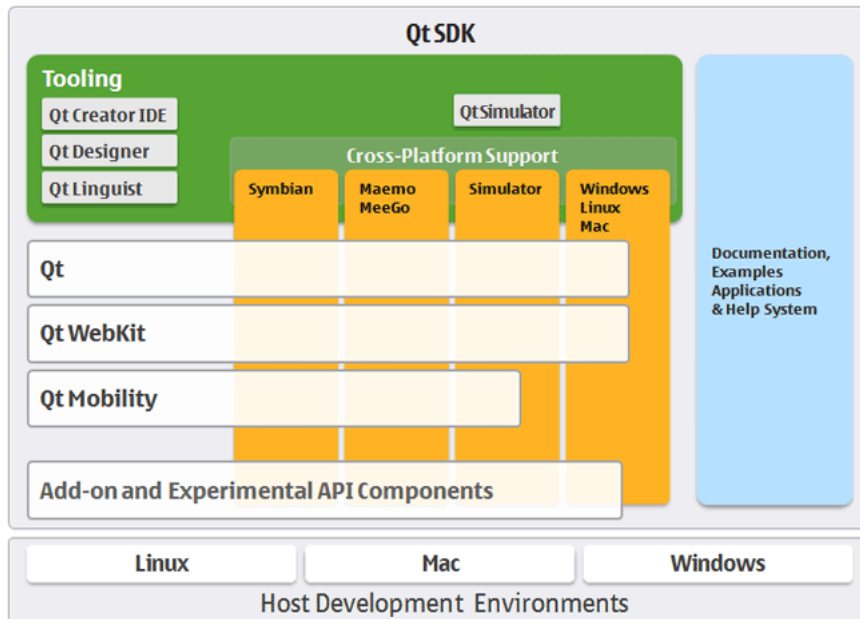
Der Sourcecode soll unter der GPL [6] zu Verfügung stehen, damit dem Kollaborieren nichts im Wege steht.

Keine zwei Jahre später steht das erste Release, mit einer Umbenennung von „Kool Desktop Environment“ nach „K Desktop Environment“ , vor der Tür und präsentiert eine integrierte Arbeitsumgebung für Betriebssysteme aus der Unix Welt. Im Zweijahresrhythmus erscheinen KDE2 und KDE3 und erfreuen sich großer Beliebtheit unter den Benutzern von Linux.

2008 erscheint KDE4 mit zweierlei großen Umstellungen. Zum einen wird von nun an Qt4 genutzt und das technische Fundament architektonisch flexibler und modularer aufgebaut.

## 4 Qt

Qt ist ein plattformübergreifendes Framework, zur Entwicklung von Applikationen, das seit 1994 entwickelt in der Schmiede von Trolltech ,die nach Übernahme von Nokia im Jahr 2008 “Qt Development Frameworks” heißt, entwickelt wird.[5] Qt bietet neben den Bibliotheken auch eine Entwicklungsumgebung, sowie einen Designer für Graphische Benutzeroberflächen und weitere Tools und Addons an. Nach dem Motto “Code Less. Create More. Deploy Everywhere.“ kann man in Qt geschriebene Applikationen auf verschiedenen Systemen zum Laufen bringen, ohne den Sourcecode zu ändern.[26]



## 4.1 Qt Module

Von Anfang an ist das Framework modular aufgebaut, wobei inzwischen, seit Qt4, es fragmentierter ist und effizientes Programmieren von kleineren und schnelleren Programmen ermöglicht. Im Folgenden sind einige Module aufgeführt.[22]

**QtCore** nicht-grafische Klassen, die von anderen Modulen verwendet werden.

**QtGui** Komponenten für die grafischen Benutzeroberflächen

**QtNetwork** Klassen für Netzwerk-Programmierung

**QtWebKit** Klassen für das Darstellen und Bearbeiten von Web Inhalten.

**QtOpenGL** Klassen mit der Anbindung zu OpenGL

**QtDBus** Klassen für die Unix-Entwickler für "Inter-Process Communication" mit DBus

Weitere Module bieten auch die Möglichkeit zur Kommunikation mit Datenbanken, XML-Dateien, SVG-Bildern und JavaScripten oder Kompatibilitätsklassen zu Qt3.

## 4.2 KDE Free Qt Foundation

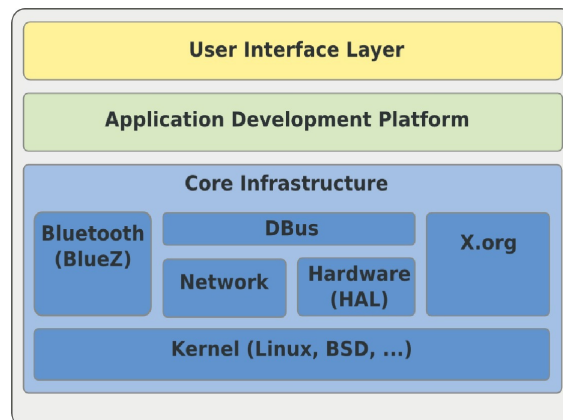
Auf Grund der Abhängigkeit von KDE zu Qt und der damaligen Befürchtung bei einer Insolvenz oder Übernahme von Trolltech, mit der möglichen Konsequenz, dass Qt unter einer unfreieren Lizenz fallen könnte, wird die KDE Free Qt Foundation

1998 gegründet. Trolltech und KDE e.V. beschließen, dass wenn die Entwicklung vom "freien" Qt nicht weitergeführt wird, Qt unter der BSD Lizenz veröffentlicht wird. Diese Vereinbarung gilt, wenn auch angepasst weiterhin nach der Übernahme von Nokia. Inzwischen ist Qt auch unter der GPL3 verfügbar.[16][23]

### 4.3 Zukunft

Mit der Kooperation von Nokia und Microsoft stellt sich die Frage inwieweit Nokia noch an Qt weiterentwickeln wird. 2012 erscheint Qt5 und diesbezüglich gibt es einige wegweisende Änderungen. Technisch gesehen möchte man die GPU<sup>2</sup> besser ausnutzen können, sowie die Erstellung von Programmen und GUI<sup>3</sup> mit QML und Javascript vereinfachen. Des Weiteren soll einfacher sein und weniger Code brauchen um Module von Qt zu nutzen. Ebenso soll in Zeiten der stärkeren Interaktion von Programmen mit dem Internet, diese so mächtig sein wie möglich. Politisch steht ein großer Umbruch bevor. Qt5 soll im Gegensatz zu Qt4, welches hauptsächlich innerhalb von Trolltech und Nokia entwickelt wird, demokratisch entwickelt werden. Hierzu wird Qt5 vollkommen Open Source, und jeder hat die Möglichkeit an Qt5 direkt mitzuentwickeln. Bezeichnet wird dieses Modell zu Entwicklung als "Open Governance". Dies bedeutet jedoch nicht, dass die Unterstützung von Nokia eingestellt wird. [20] [27]

## 5 Linux und KDE



An dem oben aufgeführten Bild[24] erkennt man die grobe Struktur eines Unix-Desktops für Endbenutzer mit KDE4. Die "Core Infrastruktur" bildet auf Basis

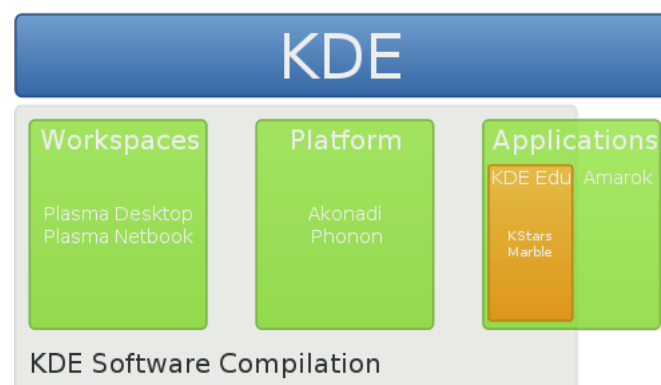
<sup>2</sup> Graphics Processing Unit - Grafikprozessor

<sup>3</sup> Graphical User Interface

des Kernels ein "Grundgerüst" für KDE. Auf dem Kernel, der die Kommunikation zur Hardware ermöglicht, sitzen weitere Komponenten, wie das X Window System "X.org", der Kommunikationsdienst für Prozesse "DBus" und "HAL" ("hardware abstraction layer"). Zusätzlich existieren weitere Komponenten, auf die hier nicht weiter eingegangen wird. Basierend auf diesem "Grundgerüst" werden die Bibliotheken geladen (Application Development Platform), die das Ausführen des Desktops, in diesem Falle KDE, und dessen Anwendungen (User Interface Layer) ermöglichen. Äquivalent dazu lassen sich zum Beispiel auch GTK+ Bibliotheken laden und anstatt KDE GNOME benutzen. GNOME ist eine weitere Arbeitsumgebung mit ähnlichem Funktionsumfang wie KDE. [25] Daran erkennt man, dass der Linux-Desktop modular aufgebaut ist und die einzelnen Komponenten ersetzbar sind.

## 6 KDE SC

Die KDE Software Compilation, weiterhin mit KDE SC abgekürzt, ist kurz gefasst eine Zusammenstellung von Bibliotheken, Programmen und Arbeitsflächen, die in einem Release-Zyklus synchron in einer Version herausgebracht werden. Letztendlich kann man KDE SC in drei "Säulen" aufteilen. Für den normalen Nutzer ist die "Säule" Workspaces eher von Interesse, weil dort die Arbeitsflächen und deren Verhalten definiert werden. Weiterhin sind natürlich auch die Applications von Bedeutung, in denen eine gewisse Anzahl an Applications, also Programme, zur KDE SC gehören und weitere von Anderen angeboten werden. Für Entwickler ist die KDE Plattform von Nutzern, in der die Funktionalitäten bereit gestellt werden.



### 6.1 KDE Workspaces

KDE SC bietet derzeit offiziell zwei verschiedene Arbeitsflächen an. Zum einen die Desktop Umgebung und zum anderen eine Arbeitsfläche für Netbooks. Beide vermitteln eine moderne Benutzung des Computers.



### **6.1.1 Plasma Desktop**

Plasma Desktop ist von Anfang an entwickelt worden ist nach Außen hin das Aushängeschild für KDE. Es bietet die verschiedensten Programme und Einstellungsmöglichkeiten an um dem Nutzer ein eine moderne Desktop Erfahrung zu bieten. Es bietet Widgets,also keine Tools, an ,die auf dem Desktop platziert werden können.Weiterhin existieren z.B. 3D -Effekte die mit Gestiken gesteuert werden können und somit den Arbeitsfluss erleichtern können. Praktisch sind auch die Aktivitäten, welche es erlauben verschiedene Konfigurationen der Arbeitsfläche zu speichern und diese im laufenden Betrieb zu wechseln. Beispielsweise kann man eine Aktivität für Zuhause und eine für die Arbeit erstellen, diese jeweils nach den Bedürfnissen konfigurieren und immer wieder zwischen ihnen wechseln. [17]

### **6.1.2 Plasma Netbook**

Plasma Netbook ist eine Arbeitsfläche für Geräte mit einer geringeren Bildschirmgröße beziehungsweise einer geringeren Auflösung gedacht. Die Oberfläche ist so angepasst,dass man ohne Mühe Programme starten kann und ebenso schnell nach Informationen, sei es bei Wikipedia oder im eigenen Adressbuch, suchen kann. Ebenso wie der Plasma Desktop, lässt sich auch Plasma Netbook und dessen Konfiguration um weitere Widgets und Programmen den Bedürfnissen erweitern.Mögich ist ein Widget,das eine bestimmte Website anzeigt,damit man sie immer im Blick hat. [17]

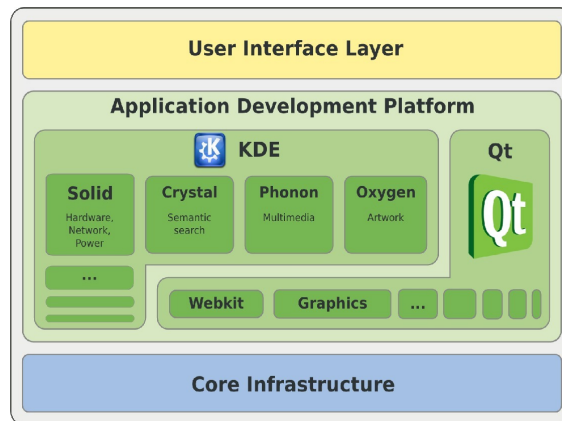
### **6.1.3 Plasma Mobile**

Plasma Mobile ist eine weitere Plasma Arbeitsfläche, die speziell für Smartphones gedacht ist. Es wird das Konzept der Aktivitäten erweitert,die jeweils für Telefonieren, das Surfen und weitere Funktionalitäten zuständig sind. Anfang 2010 wurde das Projekt gestartet,bietet allerdings noch keine offzielles und stabiles Release an. [21]

## **6.2 KDE Platform**

Softwaretechnisch interessant ist die Entwicklungsplattform von KDE SC. Mit der Zeit bemerkt man natürlich, dass das gemeinsame Entwickeln von Vorteil ist,weil das Rad neu zu erfinden nicht die Kunst des Programmierens ist. Wenn man die Funktionen der Programme ansieht,stellt man fest,dass bestimmte Teile des Codes eine Schnittmenge bilden, die, auch wenn die Implementation nicht unbedingt identisch

ist, extrahiert werden kann. Wenn weitere Programme diese Funktion nutzen möchten, so können sie auf die bisherige Implementation zurückgreifen, anstatt diese selbst zu implementieren. Von Vorteil ist es außerdem, dass Verbesserungen beziehungsweise Bugfixes allen Programmen zukommen. KDE verfügt über einige solcher extrahierten Technologien, die es ermöglichen Programme effizient und schnell zu entwickeln. Im Folgenden ist ein Teil der Plattform beschrieben. [13]



### 6.2.1 Plasma

Plasma ist der visuelle Grundpfeiler von KDE4. Es ersetzt KDesktop aus dem vorherigen KDE und ist mit dem "Qt Graphics View Framework" implementiert. Sie bietet einen Desktop mit einer Widget-Engine, mit der Widgets, sogenannte Plasma-Widgets, auf dem Desktop platziert werden können. Es können beispielshalber kleine Browserfenster, egal ob Web oder Datei, sein oder "Monitore", die die Auslastung der CPU überwachen. Ebenso werden auch "Google Gadgets" und Widgets auf Basis von SuperKaramba unterstützt.

Die zentralen Rollen spielen folgende Klassen in der Bibliothek "libplasma" [14]

**Corona** erweitert QGraphicsScene und ermöglicht das Hinzufügen von Applets.

**Widget** erweitert QGraphicsItem und erlaubt einfachste Elemente auf dem Desktop.

**Applet** erweitert Widget und beinhaltet Funktionen, wie z.B. Darstellen oder Abspielen einer lokalen Datei.

**DataEngine** ermöglicht das Laden von lokalen Dateien.

### 6.2.2 Oxygen

Ein weiterer grafischer Grundpfeiler von KDE SC ist Oxygen. Es ist eine Sammlung von vektorbasierten Icons, die KDE4 das "Look and Feel" geben. Ohne Weiteres

lässt sich die Sammlung ergänzen oder gar ersetzen,so dass im gesamten System, die gewünschten Icons zu finden sind.

### 6.2.3 Solid

Solid ermöglicht die reibungslose Kommunikation zwischen Programm und Hardware. Es befähigt einen zwar nicht direkt die Hardware über Solid anzusprechen, doch Solid zeigt welche Wege existieren, Bibliotheken oder Stacks, um mit der Hardware interagieren zu können. Ziel von Solid ist es wie ein Vermittler zwischen Frontend und Backend zu agieren um plattformunabhängig entwickeln zu können, was den Vorteil hat,dass man bei verschiedenen Plattformen, wie Linux und Windows, nicht die Anwendung anpassen muss, da diese nur das Frontend nutzen und das Backend sich um die Kommunikation mit der Hardware kümmert.[15]

**Solid::DeviceNotifier** verschafft einem die Informationen über die verfügbare Hardware.

**Solid::DeviceManager** beinhaltet die verfügbare Hardware des Systems.

**Solid::Device** repräsentiert ein Gerät und beinhaltet die spezifischen Interfaces.

**Solid::DeviceInterface** stellt eine Funktion eines bestimmten Gerätes dar.

### 6.2.4 Phonon

Ähnlich zu Solid arbeitet Phonon auch als Vermittler. Phonon ist eine plattformunabhängige Multimedia-API, die zwischen Programm und Frameworks, wie gstreamer und xine. [18] Als Entwickler muss man sich nicht mehr um die Konfiguration der Soundkarte. Denn diese Einstellungen werden einmal mit Hilfe von Solid vorgenommen.

### 6.2.5 Nepomuk

Der semantische Desktop ist schon länger ein Thema und wird mit Nepomuk (Networked Environment for Personalized, Ontology-based Management of Unified Knowledge) in KDE4 realisiert. Er "zielt darauf ab die künstlichen Grenzen zwischen den Informationen zu entfernen, um eine dynamische Klassifikation, Organisation und Präsentation der Daten für den Anwender zu ermöglichen.“[18] Sein Ziel ist es mit allen möglichen Metadaten umgehen zu können um dem Benutzer allumfassende Suche zu ermöglichen.[19]

### 6.2.6 Akonadi

Akonadi ist der "Datenverwaltungsdienst für alle PIM (Personal Information Manager) Daten".[18] Er ermöglicht es, einen zentralen Zugriff auf E-Mails, Kalendereinträge oder Kontakte und deren Adressen zu haben, wodurch die Programme nicht die selbe Funktion getrennt implementieren müssen. Ein Verwendungszweck ist die Verknüpfung von Akonadi mit externen Quellen, wie Google Kalender, womit für Kontakt, einem PIM-Programm, diese Einträge über Akonadi zur Verfügung stehen. Realisiert wird Akonadi nach dem Prinzip von "divide and conquer" mit drei Kategorien von Komponenten.[10]

**Akonadi Server** besteht aus zwei Rollen. Zum einen kontrolliert "control", implementiert als `akonadi_control`, alle Komponenten von Akonadi. Er startet den "cache", implementiert als `akonadiserver`, der die Daten bereithält und zwischenspeichert.

**Akonadi Agents** sind Komponenten die Prozesse beschreiben, die sich zum Beispiel um das Beschaffen der Daten kümmern. Akonadi Resources befüllen sozusagen den "cache" mit den Daten von anderen Quellen, wie Google Kalender.

**Akonadi Applications** sind Komponenten, die aus Sicht der Anwender auf die Daten aus dem "cache" zugreifen.

### 6.2.7 Decibel

Kommunikation ist ein großer Bestandteil des alltäglichen Lebens und der Arbeit. Decibel knüpft hier an und möchte ein Framework zur Verfügung stellen, das die verschiedenen Protokolle für Chatten, Audio und Videokonferenzen versteht. Es soll Entwicklern den Einbau von Kommunikationsmitteln in ihre Produkte vereinfachen. [12]

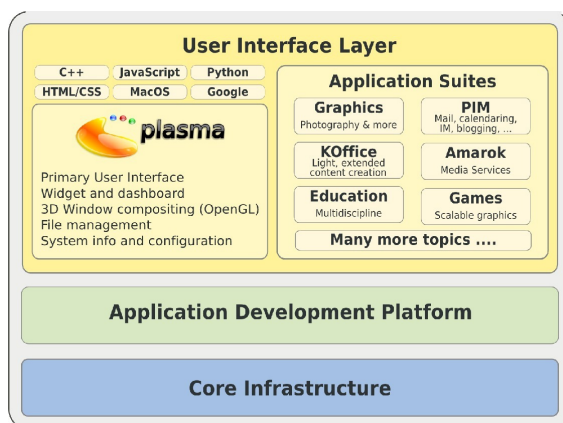
- Benutzt die TAPIOCa Bibliothek um die Telepathy Spezifikation zu realisieren
- Decibel daemon, "Houston"<sup>4</sup>
- Purer Qt4 Code, ohne Abhängigkeit zu anderen KDE Bibliotheken

---

<sup>4</sup> Ein Daemon ist ein im Hintergrund laufender Prozess in unixoiden Systemen, der bestimmte Dienste zur Verfügung stellt.

## 6.3 KDE Applications

Ein großer Bestandteil von KDE SC sind die Programme, welche in nahtlos in das moderne Erscheinungsbild der Arbeitsfläche einfügen. Sie sind in verschiedene Pakete unterteilt und bieten eine komplette Palette an Funktionalitäten.



**KDE-Plasma-Addons** bieten verschiedene "Looks" und Widgets für die Arbeitsflächen an.

**KDE-Network** stellt z.B. Instant Messenger oder Downloadmanager zur Verfügung.

**KDE-Graphics**, worunter Dokument- und Bildbetrachter wie Okular und Gwenview fallen.

**KDE-Multimedia** beinhalten Video und Musikplayer.

**KDE-Pim** enthält Tools für Mails, Kontaktadressen, Termine, Kalender usw.

Des Weiteren gibt es Pakete, die z.B. speziell für das Lernen gedacht oder reich an Spielen sind. Andere liefern hingegen Tools für das Administrieren des Systems, Bindings für andere Programmiersprachen, wie Python, Java, Perl usw. und natürlich auch Software Development Kit aus, die das Entwickeln erleichtern. Nicht zu unterschätzen ist das Paket für Office und Produktivität. Es bietet alle Funktionen, die für das normale Benutzen eines Computers als Arbeitsgerät notwendig sind. [11]

## 7 Aussicht

Wie man am Kapitel über die KDE Plattform sehen kann, ist KDE SC recht modular aufgebaut und bietet eine Vielzahl an Frameworks an. Softwaretechnisch hat man bei der Entwicklung von KDE SC bedacht, alles soweit zu modularisieren.

In naher Zukunft wird sich eine neuer Rewrite von KDE, wie es in der Evolution von KDE3 auf KDE4 geschehen ist, nicht ereignen. Qt5 wird zwar wahrscheinlich 2012 erscheinen, doch keine gravierenden Änderungen erfordern, falls ein zeitnaher Umstieg von Qt4 auf Qt5 sich in KDE SC durchsetzt.[20] Bezüglich Plasma steckt eine neue Bibliothek in den Startlöchern. Libplasma2 soll QGraphicsView von QML ersetzen.[28]

KDE SC ist, auch wenn es größtenteils von der Community entwickelt wird, ein achtbares Konkurrenzprodukt zu den anderen Arbeitsoberflächen, wie Windows und MacOSX und bietet ein Vielerlei an Funktionen, die die normalen Benutzer begeistern können, und softwaretechnisch soviel Potenzial an, das für Entwickler äußerst interessant sein sollte. Im Gegensatz zum Server-Bereich, in dem Linux immer noch marktführend ist, ist dies auf dem Desktop-Bereich nicht der Fall. Ebenso kann man Linux auf dem Mobilien Markt einen durchschlagenden Erfolg nachsagen, wenn man die Verteilung von Symbian und Android im Gegensatz zu Windows ansieht.[3] Auch in der Automobil-Welt scheint es einen Trend zu Linux zu geben, wie man an dem neuesten Mitglied Toyota der Linux Foundation sehen kann.[9] Einzig der Desktop-Bereich ist nicht erfolgreich. Dies mag wohl daran liegen, dass man kaum Computer oder Notebooks bekommen kann, auf denen nicht Windows vorinstalliert ist. Zugleich gibt es noch wenige technisch Unversierte, die sich an das Thema Linux trauen, da dies als umständlich verschrien ist<sup>5</sup>. Mittlerweile kann man dies jedoch zum größten Teil, je nach Distribution, als widerlegt ansehen, da die Installation leicht, wenn nicht sogar leichter, als die von Windows ist. Auch wenn sich ein großer Erfolg, der Windows in die Schranken weist, im Desktop-Bereich nicht abzeichnet, so bietet die Modularität bzw. das Design von KDE, ihm die Möglichkeit sich in anderen Bereichen zu etablieren. Vielversprechend ist "Meego" als Betriebssystem für Smartphones, Tablets, Entertainment Systeme usw., auf der Basis auch KDE auch ausgeführt werden kann. Doch leider nur bis zu dem Zeitpunkt der Kooperation von Nokia und Windows, denn es ist nun ungewiss in Wieweit Meego sich noch gegen Googles Android und Apples iOS behaupten kann. Ebenfalls lassen sich Qt und damit auch KDE Bibliotheken auf Android [2] und Symbian [1] ausführen. Dies ermöglicht die Wiederverwendung von Programmen, die sowohl für den Desktop, als auch für das Smartphone geeignet sind. Als Beispiel könnte man die Plasmoids anführen, die über das Netzwerk verteilt werden können und somit eventuell den Sprung auf das Smartphone schaffen? Ob dies auch die Bekanntheit von KDE-Desktop bzw. Linux-Desktop steigert, lässt sich nicht beantworten. Zu wünschen wäre dies, denn das Potenzial ist auf jeden Fall vorhanden. . .

---

<sup>5</sup> Stichwort: Kompilieren

## Literatur

- [1] allaboutsymbian.com. *Nokia accelerates Qt focus, continuous improvement for Symbian*. URL: [http://www.allaboutsymbian.com/news/item/12215\\_Nokia\\_accelerates\\_Qt\\_focus\\_con.php](http://www.allaboutsymbian.com/news/item/12215_Nokia_accelerates_Qt_focus_con.php).
- [2] aportale. *Bringing Qt applications to Android*. URL: <http://labs.qt.nokia.com/2011/02/28/necessitas/>.
- [3] comscore.com. *Smartphone Platform Market Share*. URL: [http://www.comscore.com/Press\\_Events/Press\\_Releases/2011/7/comScore\\_Reports\\_May\\_2011\\_U.S.\\_Mobile\\_Subscriber\\_Market\\_Share](http://www.comscore.com/Press_Events/Press_Releases/2011/7/comScore_Reports_May_2011_U.S._Mobile_Subscriber_Market_Share).
- [4] Matthias Ettrich. *New Project: Kool Desktop Environment (KDE)*. URL: <http://www.kde.org/announcements/announcement.php>.
- [5] Qt Development Frameworks. *Who we are*. URL: <http://qt.nokia.com/about/who-we-are>.
- [6] GNU. *GPL*. URL: <http://www.gnu.org/licenses/gpl.html>.
- [7] The Open Group. *Common Desktop Environment*. URL: <http://www.opengroup.org/cde/>.
- [8] GI Gesellschaft für Informatik. *Handbuch der Software-Architektur*. URL: <http://www.handbuch-softwarearchitektur.de/>.
- [9] inuxfoundation.org. *Toyota Joins Linux Foundation*. URL: <http://www.linuxfoundation.org/news-media/announcements/2011/07/toyota-joins-linux-foundation>.
- [10] KDE.org. *Akonadi*. URL: <http://techbase.kde.org/Development/Architecture/KDE4/Akonadi>.
- [11] KDE.org. *Applications*. URL: <http://userbase.kde.org/Applications>.
- [12] KDE.org. *Decibel*. URL: <http://techbase.kde.org/Development/Architecture/KDE4/Decibel>.
- [13] KDE.org. *Development Platform*. URL: <http://www.kde.org/developerplatform/>.
- [14] KDE.org. *Development/Architecture/KDE4/Plasma*. URL: <http://techbase.kde.org/Development/Architecture/KDE4/Plasma>.
- [15] KDE.org. *Development/Architecture/KDE4/Solid*. URL: <http://techbase.kde.org/Development/Architecture/KDE4/Solid>.

- [16] kde.org. *KDE Free Qt Foundation*. URL: <http://www.kde.org/community/whatis/kde/kdefreeqtfoundation.php>.
- [17] KDE.org. *KDE SC Workspaces*. URL: <http://www.kde.org/workspaces/>.
- [18] KDE.org. *KDE Userbase Glossary*. URL: <http://userbase.kde.org/Glossary/de>.
- [19] KDE.org. *Nepomuk*. URL: <http://nepomuk.kde.org/>.
- [20] Lars Knoll. *Qt5*. URL: <http://labs.qt.nokia.com/2011/05/09/thoughts-about-qt-5/>.
- [21] Alexis Menard. *Tokamak 4, the KDE Plasma meeting*. URL: <http://labs.qt.nokia.com/2010/02/28/tokamak-4-the-kde-plasma-meeting/>.
- [22] Qt Module. *Qt Development Frameworks*. URL: <http://doc.qt.nokia.com/latest/modules.html>.
- [23] Nokia. *KDE Free Qt Foundation*. URL: <http://qt.nokia.com/about/news/archive/00000004>.
- [24] Jos Poortvliet. *Generic KDE 4 talk*. URL: [http://www.kde.org/kdeslides/KDE4talk/generic\\_KDE4talk.odp](http://www.kde.org/kdeslides/KDE4talk/generic_KDE4talk.odp).
- [25] The GNOME Project. *GNOME*. URL: <http://www.gnome.org/>.
- [26] Qt. *Nokia*. URL: <http://qt.nokia.com/>.
- [27] Cornelius Schumacher. *Qt and the Future of KDE*. URL: <http://dot.kde.org/2011/03/03/qt-and-future-kde>.
- [28] Aaron J. Seigo. *libplasma2*. URL: <http://aseigo.blogspot.com/2011/05/libplasma2.html>.
- [29] Gernot Starke und Peter Hruschka. *Software-Architektur kompakt - angemessen und zielorientiert*. 1. Aufl. Heidelberg: Spektrum Akademischer Verlag, 2011. ISBN: 978-3-8274-2093-0.