

TECHNISCHE UNIVERSITÄT ILMENAU
Rechnerarchitekturen und Eingebettete Systeme

Reviewdokument zur Implementierungsphase
RION Package-Manager

vorgelegt von:	P. Augustin, J. Skopp, C. Juin, G. Lehmann, A. Schmidt, R. Schöne
eingereicht am:	22.06.2022
Fachgebiet:	Rechnerarchitekturen und Eingebettete Systeme Institut für Mikroelektronik- und Mechatronik-System
Betreuer:	Georg Gläser, Andreas Becher
Seminarleiter	Prof. Armin Zimmermann

Contents

1	Einleitung	2
1.1	Zweck des Systems	2
1.2	Entwurfsziele	2
1.3	Überblick	3
2	Grobentwurf	4
3	Realisierter Entwurf	5
3.1	altes Klassendiagramm	5
3.2	neues Klassendiagramm	6
3.3	Sequenzdiagramm	7
	Abkürzungsverzeichnis und Formelzeichen	10

Review Documentation

1 Einleitung

1.1 Zweck des Systems

Aufgabe ist die Erstellung eines Packagemanagers zur Suche, Installation, Aktualisierung und Verwaltung von PDKs.

1.2 Entwurfsziele

Der Packagemanager sollte in Python geschrieben werden.

Zuzüglich der oben genannten Ziele, sollte der Packagemanager auch das Management von Abhängigkeiten, das Verwalten von virtuellen Umgebungen, sowie das Erkennen von bereits installierten Paketen ermöglichen. Zudem sollten auch zu jedem Paket ein Informationstext in Form von bereits vorliegenden Textdateien angezeigt werden können.

Serverseitig soll ein Programm die verfügbaren Pakete verwalten können. In einer Datenbank sollen der Name des Paketes, noch festzulegende Metadaten und Abhängigkeiten gespeichert sein. Außerdem soll der Server dem Packagemanager die Pakete bereit stellen.

Client und Server zusammen sollen über eine Rechteverwaltung bzw. Authentifizierung verfügen, welche es verschiedenen Nutzern ermöglicht, auf verschiedene Pakete zuzugreifen.

1.3 Überblick

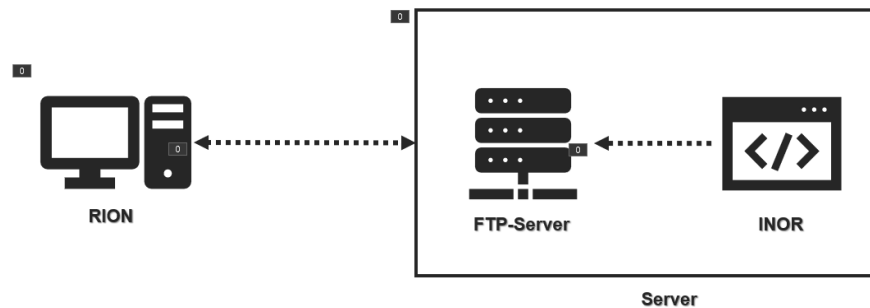
Implementiert wurden zum jetzigen Stand folgende Punkte:

- Entfernen von Paketen
- Suchen von Paketen
- Nutzerauthentifizierung
- Übertragung von Paketen vom Server zum Client
- Verwalten von virtuellen Umgebungen
- Installation von Paketen
- Konfigurationsdatei

Noch nicht implementiert sind:

- Abhängigkeitenverwaltung
- Programm zur Verwaltung der auf dem Server verfügbaren Pakete
- Erkennen von bereits installierten Paketen
- Aktualisierungen

2 Grobentwurf



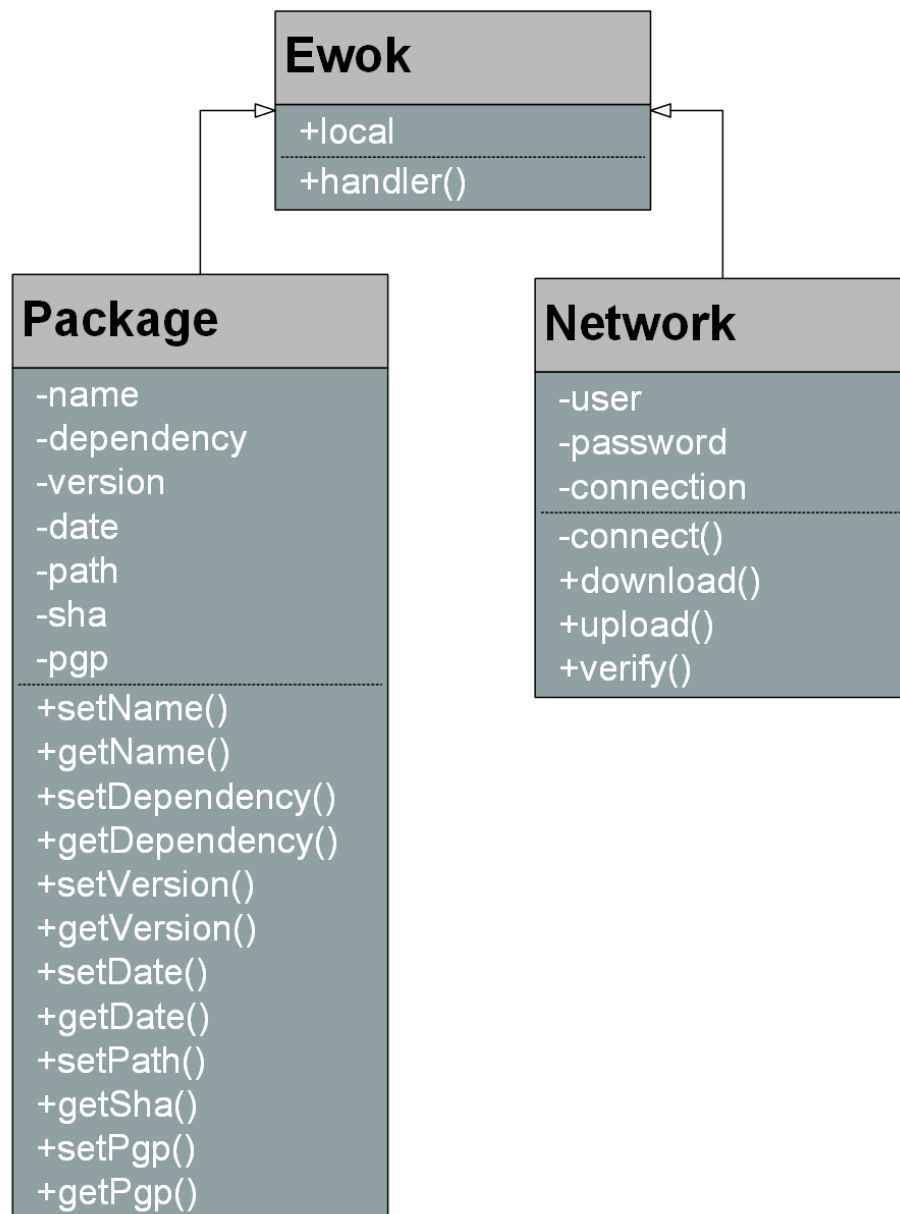
Das Programm arbeitet prinzipiell mit drei Komponenten. Zum einen gibt es den Packagemanager RION. Dieser läuft auf dem jeweiligen Endgerät (Client), der Pakete sucht, installiert, aktualisiert, entfernt und verwaltet. Hierzu werden mehrere Datenbanken verwendet. Darunter eine Datenbank, die eine Liste aller installierten Pakete enthält und eine weitere, mit einer Liste der Namen aller verfügbaren Pakete, sowie deren Metadaten und die Namen der notwendigen Abhängigkeiten.

Die Pakete, sowie die zweite genannte Datenbank, können von einem FTP-Server über eine verschlüsselte ftps-Verbindung abgerufen werden. Hierzu wird das Programm pyftplib, welches unter anderem virtuelle Nutzer zur Verfügung stellt, die zum Rechtemanagement verwendet werden. Dieses Programm wurde nicht von uns entwickelt.

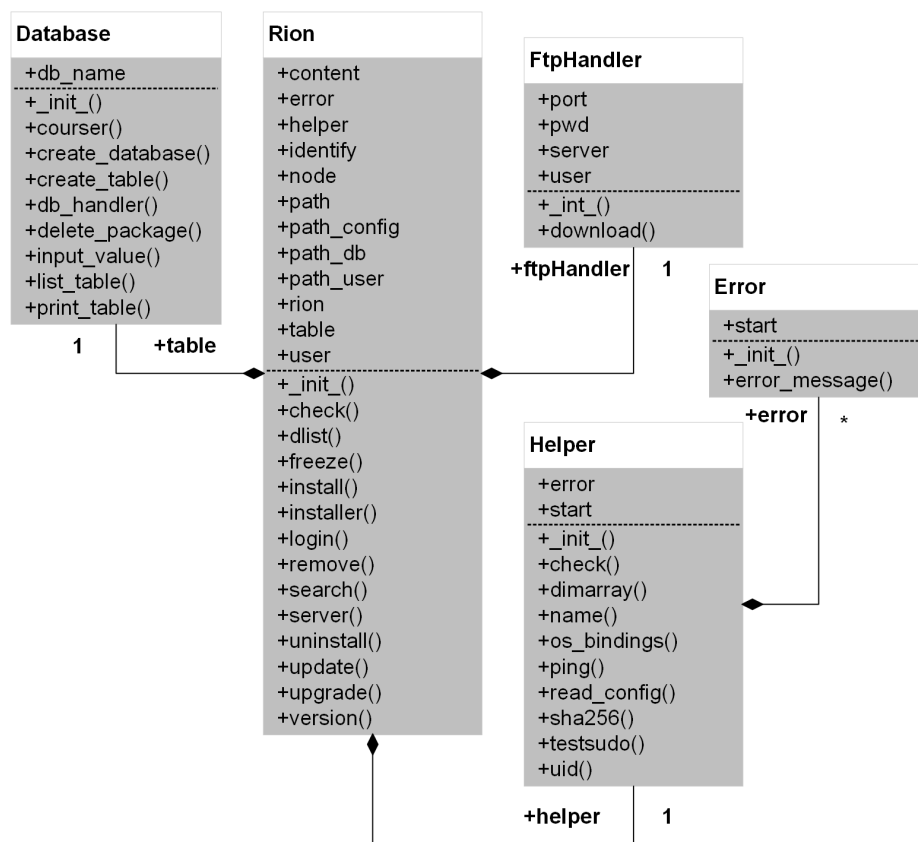
Die Daten für den FTP-Server, sowie die Konfiguration der Rechteverwaltung, werden durch die dritte Komponente INOR zur Verfügung gestellt. Durch diesen Vorgang werden einige Funktionen erleichtert bzw. automatisiert. INOR befindet sich dabei auf dem selben Server wie der FTP-Server.

3 Realisierter Entwurf

3.1 altes Klassendiagramm

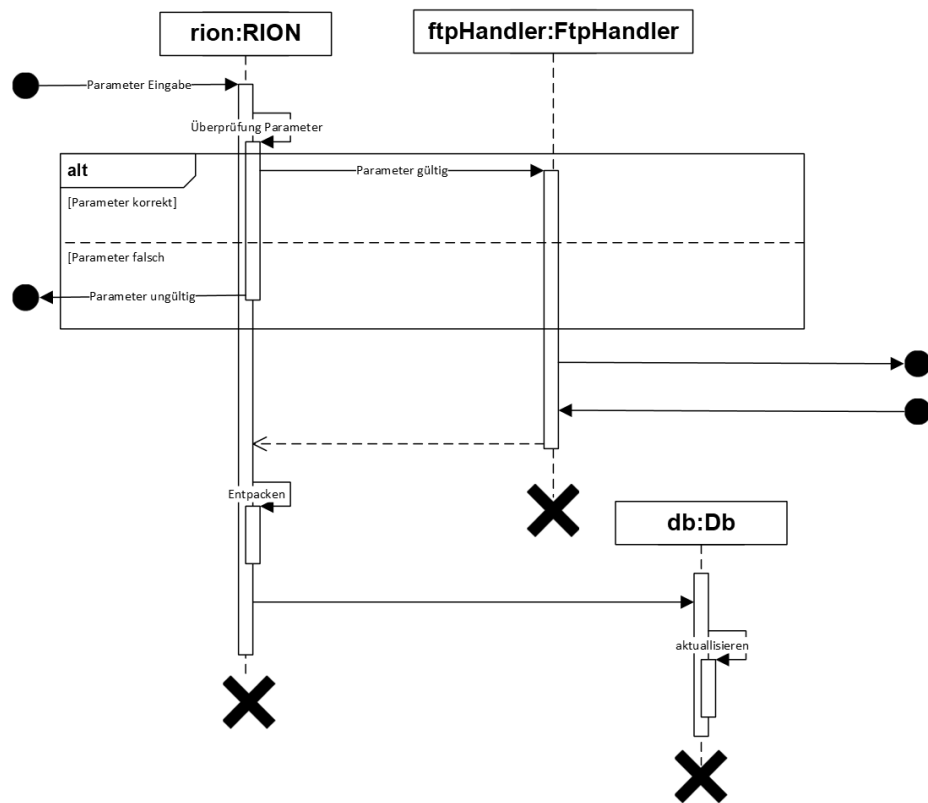


3.2 neues Klassendiagramm



In der Implementierungsphase wurde die Erstellung eines neuen Klassendiagramms, aufgrund von Unzulänglichkeiten des vorigen, notwendig.

3.3 Sequenzdiagramm



Entscheidung 1

Aufgabe: Protokoll, mit dem RION auf die Pakete des Servers zugreift

Ursprünglich war geplant, RION und INOR via Client-Server-Modell eng miteinander zu verbinden. In der eigentlichen Implementierung stellte sich jedoch heraus, dass ein klassischer Ansatz sinnvoller ist.

Inor arbeitet unabhängig von RION. Seine Aufgabe besteht allein darin, Ordner mit Paketen bzw. physischen Links auf Pakete zu befüllen. Auf diese wird dann mittels eines Protokolls zugegriffen. Hierfür gibt es primär zwei verschlüsselte Optionen:

- Option 1: https
 - Vorteile:
 - * beste Übertragungsgeschwindigkeit
 - Nachteile:
 - * Arbeitet nicht direkt auf Dateienebene
 - * Aufwendige Implementierung der Authentifikation
- Option 2: ftps
 - Vorteile
 - * Arbeitet direkt auf der Dateienebene
 - * pyftplib bietet bereits eine ausgereifte und gut getestete Implementierung zur Authentifikation in Form von virtuellen Nutzern
 - Nachteile
 - * Nicht die beste, aber immer noch eine sehr gute Geschwindigkeit

Konklusion: Aufgrund der Möglichkeit virtuelle Nutzer anzulegen, ist ftps aus unserer Sicht die für RION/INOR richtige Entscheidung.

Entscheidung 2

Aufgabe: Festlegen eines Tools zur Verwaltung von installierbaren und installierten Paketen.

- Option 1: Sqlite3
 - Vorteile
 - * native Einbindung
 - * komplexe Anfragen funktionieren einfacher
 - * leichtgewichtig
 - Nachteile
 - * Nicht ohne entsprechendes Programm lesbar
 - * Entsprechende Binarys für jede Plattform
- Option 2: json
 - Vorteile
 - * leichtgewichtiger
 - * plain text
 - Nachteile
 - * unübersichtlich bei großen Datenmengen

Konklusion: Sqlite3 mag den Nachteil haben, dass es normalerweise eine systemspezifische Binary benötigt, dies wird jedoch durch die Einbindung in Python umgangen. Dies kombiniert mit den einfachen Möglichkeiten für komplexe Anfragen, hat den Ausschlag für Sqlite3 gegeben

Abkürzungsverzeichnis und Formelzeichen

CLI	Ein Command Line Interface ist eine Schnittstelle, die es dem Nutzer erlaubt, Textbefehle an ein Programm zu übermitteln. Betriebssysteme, darunter GNU/Linux stellen hierfür eine sogenannte Shell, wie zum Beispiel Bash, Dash oder ZSH, zur Verfügung, auf welche man mit einem Terminal zugreifen kann.
INOR	Serverseitiges BackEnd bei X-FAB
Manpage	Eine Hilfeseite für installierte Programme unter Unix-artigen Systemen, auf der, für gewöhnlich, Befehle, Flags und Hinweise zur Benutzung eines Programms verzeichnet sind. Darauf zugegriffen wird mit <code>man »name«</code> aufrufen.
Pakete / Packages ..	Datenbündel, die als solches eine Funktion haben. (Hiermit sind die Pakete auf den X-FAB Servern gemeint)
PDK	Ein PDK ist eine komplexe Sammlung von technologiespezifischen Daten für den (aufwändigen und relativ teuren) Entwurf von anwendungsspezifischen integrierten Schaltkreisen
pip	Python Package-Manager: https://pypi.org/project/pip/
pyftplib	FTP-Server-Implementation für Python
PyPi	Pypi ist ein Software Sammlung wo Python Packages via pip heruntergeladen werden können
Python	Programmiersprache: https://www.python.org/
RION	Package-Manager und Schnittstelle für die Paketverwaltung zwischen X-FAB-Server und Client
X-FAB	X-FAB ist ein Anbieter für Halbleitertechnologien, welche die Fertigung als auch Designunterstützung für Kunden anbietet, die gemischt analog-digitale integrierte Schaltkreise (ICs) entwickeln. X-FAB bietet eine Fülle an unterschiedlichen Technologien für diverse und auch spezifische Anwendungsmärkte an.