

The Software–System Architecture of YEROTH–ERP–3.0

DR. XAVIER NOUMBISSI NOUNDOU

This document describes the thick–client software–system architecture of YEROTH–ERP–3.0. This document also explains the reasons for which we chose to design and implement YEROTH–ERP–3.0 as a thick–client software–system, as opposed to currently more popular web–browser–based software–system.

Contents

Contents	2
List of Figures	3
List of Tables	4
1 PROFESSOR Biography	5
2 Introduction	6
2.1 Motivation	6
2.2 Thick–Client VS Web–Browser–based Software–System Architecture	6
3 The Thick–Client Software–System Architecture of YEROTH–ERP–3.0	8
3.1 Business and user interface code deployment	8
3.2 Co–related software–systems	8
3.3 User interface	8
3.4 Number of logical layers	8
4 Conclusion	9
5 Bibliography	11

List of Figures

1.1	Portrait of Xavier.	5
2.1	2–layers logical architecture of thick–client software–system (Image copied from [sec20]).	6
2.2	4–layers logical architecture of web–browser–based software–system (Image copied from [KM06]).	7

List of Tables

2.1 Thick–client application VS Web–browser–based application.	6
4.1 YEROTH–ERP–3.0 VS. Odoo web–browser–based software–system.	9

Chapter 1

PROFESSOR Biography

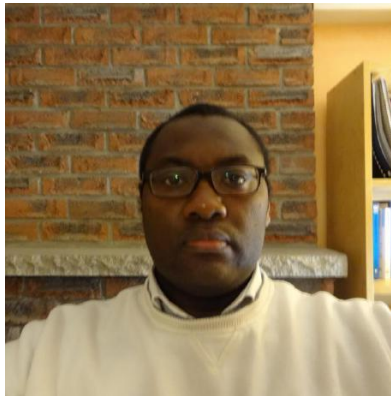


Figure 1.1: Portrait of Xavier.

DR. XAVIER NOUMBISSI NOUNDOU is a CHRISTIAN BY FAITH, Cameroonian, born on September 16 1983 in DOUALA (LITTORAL region, CAMEROON).

Xavier has a “*Diplom–Informatiker (Dipl.–Inf.)*” qualification from the **University of Bremen, Bremen, GERMANY** (May 25, 2007).

Xavier is a *PH.D. in Software Engineering* (software construction, and testing) since November 18, 2020 because of his academic research, and professional engineering contributions as follows:

1. ‘Context-Sensitive Staged Static Taint Analysis For C using LLVM’

1. source code: <http://github.com/sazzad114/saint>
2. full text (published on July 1st, 2015): http://archive.org/details/saint_201507.

2. ‘YEROTH-ERP-3.0’: <http://archive.org/details/yeroth-erp-3-0-info-english>.

Chapter 2

Introduction

2.1 Motivation

YEROTH–ERP–3.0 is an **Enterprise Resource Planing (ERP)** software–system that aims ‘effectiveness’ and ‘simplicity’, compared to other high ranked ERP software–systems (e.g.: ‘Sage Gescom i7’, ‘SAP Business One’, etc.).

We chose to design and implement YEROTH–ERP–3.0 as a thick–client software–system because of the following reasons:

1. the implementation language C++ offers much flexibility (use of macro, multiple inheritance, etc.)
2. the availability of ‘WHAT YOU SEE IS WHAT YOU GET’ (WYSIWYG) tools for fast and useful user interface design (e.g.: Qt designer [Com20], miniStudio (vxWorks) [WEI20], etc.)
3. the low number of logical software architecture layer (2) involved with the operation of a thick–client software–system, as opposed to a web–browser–based software–system (with at least a 4 layers in its logical software architecture).

2.2 Thick–Client VS Web–Browser–based Software–System Architecture

	Thick–client application ✓	Web–browser–based application
business code	all computers	application server
co–related software–systems	1 (DBMS)	at least 3 (DBMS, web / application server)
user interface	all computers (thick–client gui)	all computers (web–browser)
number of logical layers	2 (client and data)	4 (client, presentation, logic, and data)
rapid prototyping (WYSIWYG tools)	yes	very limited
software security vulnerability	low (1 programming language)	high (several programming languages)

Table 2.1: Thick–client application VS Web–browser–based application.

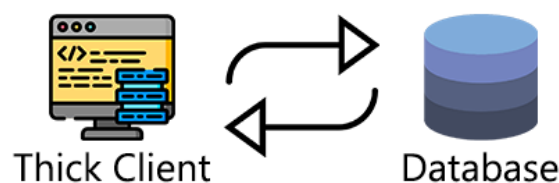


Figure 2.1: 2–layers logical architecture of thick–client software–system (Image copied from [sec20]).

Figure 2.1 illustrates an example of a thick–client software–system with a 2–layers logical architecture.

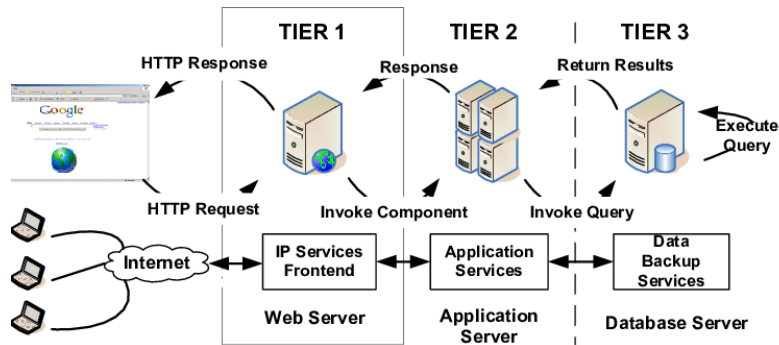


Figure 2.2: 4–layers logical architecture of web–browser–based software–system (Image copied from [KM06]).

Figure 2.2 illustrates an example of a web–browser–based software–system with a 3–layers logical architecture.

Table 2.1 compares thick–client software–systems against web–browser–based software–systems.

Chapter 3

The Thick–Client Software–System Architecture of YEROTH–ERP–3.0

3.1 Business and user interface code deployment

Table 2.1 depicts the issue of business and user interface code deployment on all computers participating in the functioning of YEROTH–ERP–3.0, as a software–system for a user.

We tackle the problem of automatic deployment of business and user interface code on all user computers by using the ‘apt upgrade’ software–system on ‘Debian–Linux’.

3.2 Co–related software–systems

3.3 User interface

3.4 Number of logical layers

Chapter 4

Conclusion

	YEROTH–ERP–3.0	Odoo (Web-browser-based application)
libraries & programs	lxml, etc.	Log4j, etc.
business code	C++	Java
DBMS	MySQL	MySQL
web-server		any (e.g.: Tomcat)
application server		e.g.: JBoss

Table 4.1: YEROTH–ERP–3.0 VS. Odoo web-browser-based software-system.

YEROTH–ERP–3.0 has a thick-client software-system architecture because we found thick-client software-system architectures simpler than web-browser-based software-system architectures.

Thick-client software-system architecture are simpler because it requires less layers in its logical software-system architecture. Table 2.1 illustrates a thick-client software-system requires less layers in its logical software-system architecture than a web-browser-based software-system.

A web-browser-based software-system potentially entails more present software security vulnerabilities because its implementation requires to use at least 2 different programming languages, and frameworks in combination.

A web-browser-based software-system architecture has more drawbacks as follows:

1. it requires at least 3 other software-system applications, *apart from the ones normally required by developer software-system itself, for instance libraries (e.g.: Log4j), to fully operate (e.g.: DBMS, web server, application server.)*.

Table 4.1 depicts this situation based on the open source ERP software-system Odoo.

2. A web-browser-based software-system requires at least 4 layers in its logical system architecture (e.g.: client, presentation, logic, and data layers).
3. A web-browser-based software-system potentially possesses more software security vulnerabilities because its implementation requires to use at least 2 different programming languages, and frameworks in combination.

YEROTH–ERP–3.0 Software–System Product Sheet

YEROTH–ERP–3.0 is an **ERP software–system** with 6 **user roles, and types**:

1. « Administrator »
2. « Business manager »
3. « Cashier »
4. « Seller »
5. « Stock manager »
6. « Storekeeper ».

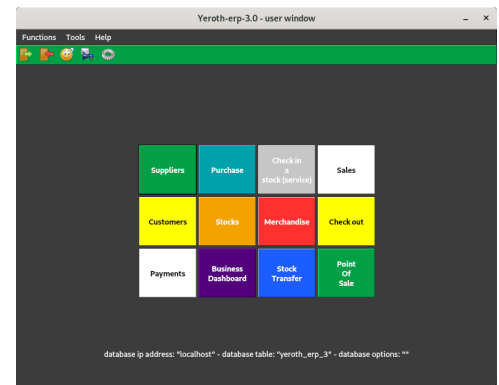
YEROTH–ERP–3.0 **features**:

1. alerts over stock quantity, and, time period
2. business dashboard
3. customer relationship management (CRM)
4. sale management (e.g. point–of–sale)
5. stock management (e.g. check in)
6. user, and role administration.

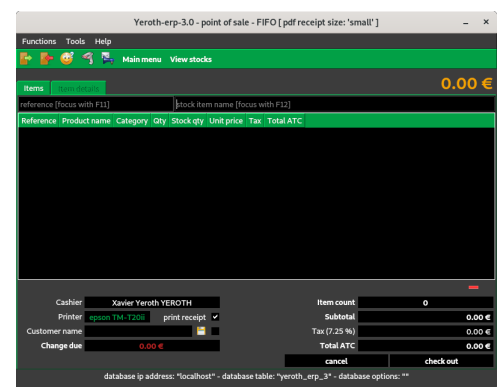
YEROTH–ERP–3.0 **is**:

1. easier, and, intuitiver, in its use
2. lighter, and, faster, in memory usage
3. multi sites (enabled).

YEROTH–ERP–3.0's runtime memory usage test is realized using software analysis tool **valgrind**.



Business manager's main window



Cashier's main window

OPERATIONS

Point–of–Sale Hardware

- ✓ Barcode scanner
- ✓ Thermal printer, etc.



Database Management Systems

- ✓ MySQL



Operating Systems

- ✓ Debian–Linux



Chapter 5

Bibliography

- [Com20] The Qt Company. Qt Designer Manual. <http://doc.qt.io/qt-5/qtdesigner-manual.html>, 2020. Last accessed on September 4, 2020 at 15:21.
- [KM06] Taeho Kgil and Trevor Mudge. Flashcache: A nand flash memory file cache for low power web servers. In *Proceedings of the 2006 International Conference on Compilers, Architecture and Synthesis for Embedded Systems*, CASES '06, page 103–112, New York, NY, USA, 2006. Association for Computing Machinery.
- [sec20] securityboulevard.com. Thick Client Penetration Testing Methodology. <http://securityboulevard.com/2020/02/thick-client-penetration-testing-methodology/>, 2020. Last accessed on September 4, 2020 at 15:21.
- [WEI20] Yongming WEI. miniStudio User's Guide. <http://www.minigui.net/en/ministudio>, 2020. Last accessed on September 4, 2020 at 15:21.