The Software–System Architecture of YEROTH–ERP–3.0

Dipl.–Inf. XAVIER NOUMBISSI NOUNDOU

Abstract

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

Contents

List of Tables

List of Figures

	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)
software security vulnerability	low (1 programming language)	high (*several* programming languages)

Table 1: Thick–client application VS Web–browser–based application.

1 Developer Biography

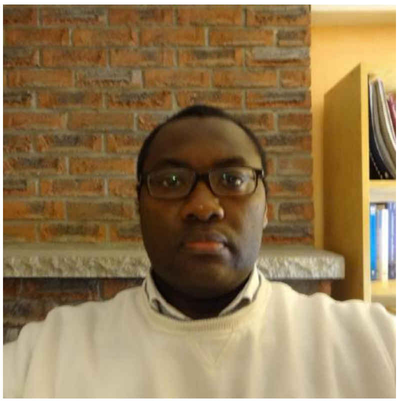

Figure 1: Portrait of Xavier.

Dipl.–Inf. XAVIER NOUMBISSI NOUNDOU is a Cameroonian born on September 16 1983 in DOUALA (LITTORAL region, CAMEROON).

Xavier is a "Diplom–Informatiker (*Dipl.-Inf.*)" of the **University of Bremen, Bremen, Bremen, GERMANY** (May 25, 2007).

2 Introduction

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, 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 computers involved with the operation of a thick–client software–system (2–layers architecture), as opposed to a web–browser–based software–system (with at least a 2–layers architecture).

3 Thick–Client VS Web–Browser

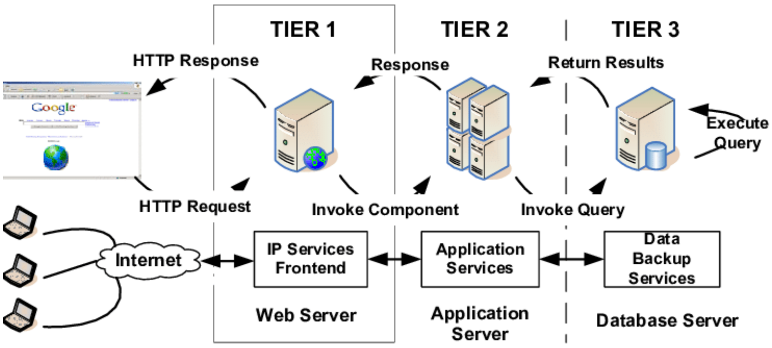

Figure 2: 4–layers logical architecture of web–browser–based software–system (copied from [KM06]).

Figure 2 illustrates an example of a web–browser–based software–system with a 3–layers logical architecture.

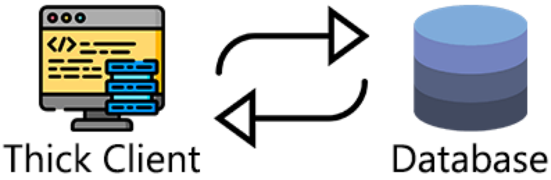

Figure 3: 2–layers logical architecture of thick–client software–system (copied from [sec20]).

Figure 3 illustrates an example of a thick–client software–system with a 2–layers logical architecture.

Table 1 compares thick–client software–systems against web–browser–based software–systems.

4 YEROTH–ERP–3.0 as a Thick–Client Software Application

4.1 Business and user interface code upgrade deployment

Table 1 depicts the issue of upgrading business and user interface code on all computers participating in the functioning of YEROTH–ERP–3.0, as a networked computer system for a potential customer company.

We tackle the problem of automatic deployment of business and user interface code on all networked computers by using the 'apt upgrade' system on 'Debian-Linux'.

4.2 Co–related software–systems

4.3 Networked architecture

4.4 Security vulnerability protection

5 Conclusion

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

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

1) it requires at least 3 co–related software–systems are required (e.g.: DBMS, web server, application server.) to fully operate.

2) A web–browser–based software–system requires at least 4 layers are required within its logical architecture (e.g.: client, presentation, logic, and data).

3) A web–browser–based software–system potentially possesses more software security vulnerabilities because its implementation requires of the use of at least 2 different programming languages, and frameworks in combination.

6 References

[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.