Architecture Specification

for

Microminer

Version 1.0 Approved

Prepared by Lynn Barnett and Victoria Potvin

University of Central Oklahoma

February 4, 2015

Contents

[1. Architectural Style: Mixed Shared Data and OO 3](#_Toc412666505)

[1.1 Shared Data 3](#_Toc412666506)

[1.2 OOT – Object Oriented Techniques 3](#_Toc412666507)

[1.3 Advantages 3](#_Toc412666508)

[1.4 Disadvantages 3](#_Toc412666509)

[1.5 Conclusions 3](#_Toc412666510)

[2. Diagrams 4](#_Toc412666511)

[2.1 Component Diagram 4](#_Toc412666512)

[2.2 Deployment Diagram 5](#_Toc412666513)

# Architectural Style: Mixed Shared Data and OO on a 3-tiered Architecture

The project will be implemented using a mix of Shared Data and OO Architecture.

## Shared Data

* The components of this architecture are processes and data.
* The connectors of this architecture are direct memory access, subprogram calls, and system I/O.

## OOT – Object Oriented Techniques

Object oriented techniques aim at encapsulating data and defining class relationships well to create easy to maintain software.

## 3-tiered Client Server Architecture

* The components of this architecture are clients and servers.
* The connectors of this architecture are networks that connect clients and servers.

## Advantages

* Shared data helps performance in the case of large datasets because the data does not have to be copied by components.
* Shared data allows for easier implementation of interactive functions.
* Information hiding and well defined relationships due to OO mean an easier to maintain piece of software.
* Client Server means that the application does not have to be installed on a computer to be used.
* 3-tiered means that the data can be stored on a different computer than the application logic.
  + This means a better balance load on the servers.
* Classes are easier to change without unintentionally affected other classes.

## Disadvantages

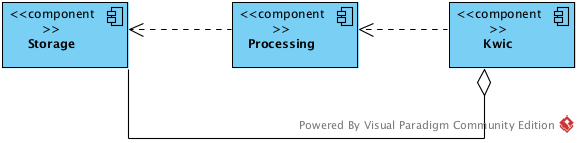
* With Shared Data architecture processing by components must be done sequentially, therefore not in parallel.
* Object oriented techniques can sometimes result in bloated and over-complicated applications.
* N-tiered architectures can have a heavier load on the network than 2-tiered architectures.

## Conclusions

Because the system is simple, there is not much need for parallel processing or worry about a bloated system. Likewise since this is not processing huge amounts of data there is little concern about overloading the network. Shared data is a good choice since we desire to create the software as a web application and in the future it will be more interactive. OOT is a good choice because our software will be easy to maintain in the future. 3-tiered is a good choice because it is most similar to the way a true search engine would be structured and allows us to (theoretically) avoid overloading one computer.

## Diagrams

## Component Diagram



## Deployment Diagram