

# The Ekans Webserver

## **Software Design Specification**

Samuel C. Payson

*last modified*  
December 22, 2010

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	The Ekans Design Mentality . . . . .	1
1.2	Understanding Figures . . . . .	1
<b>2</b>	<b>Overview</b>	<b>2</b>
2.1	The Main Modules . . . . .	2

## 1 Introduction

Ekans is a lightweight webserver designed to quickly serve static content, while allowing for dynamic content powered by a Python backend. The initial section of this document provides a concise high-level overview of the server's design, followed by more detailed sections explaining the implementation of each of the server's main modules.

### 1.1 The Ekans Design Mentality

As a large project written in C, Ekans adhere's primarily to the mindset of the procedural paradigm. This means that data and procedures are seperate components of each module. In our model, modules exist as a set of procedures which communicate with eachother through changes that they make to data.

The main function of Ekans is the processing and communication of hypertext data. Because of this focus, our design places emphasis on the movement of data around the system. Modules are seperated based on the order and nature of their interactions with data.

### 1.2 Understanding Figures

The flow-charts and figures throughout this document attempt to follow the same general guidelines for design in order to be as useful as possible. The following are a set of informal rules for the interpretation of figures. *Do not take this to be the absolute truth!*

## 2 Overview

### 2.1 The Main Modules

The two main modules of Ekans are known as *dispatch* and the *agent pool*. Dispatch is responsible for establishing connections with new clients and handing them off to agents. The agents handle individual clients one at a time, serving their requests and dealing with errors.

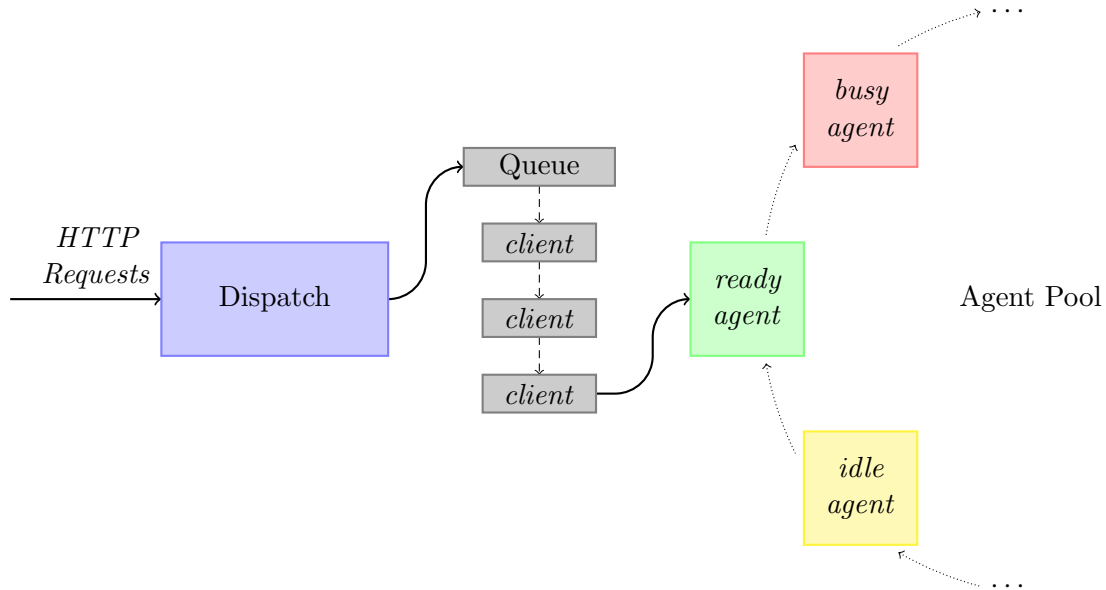


Figure 1: High level interactions of dispatch and the agent pool.