

## AskemosServer

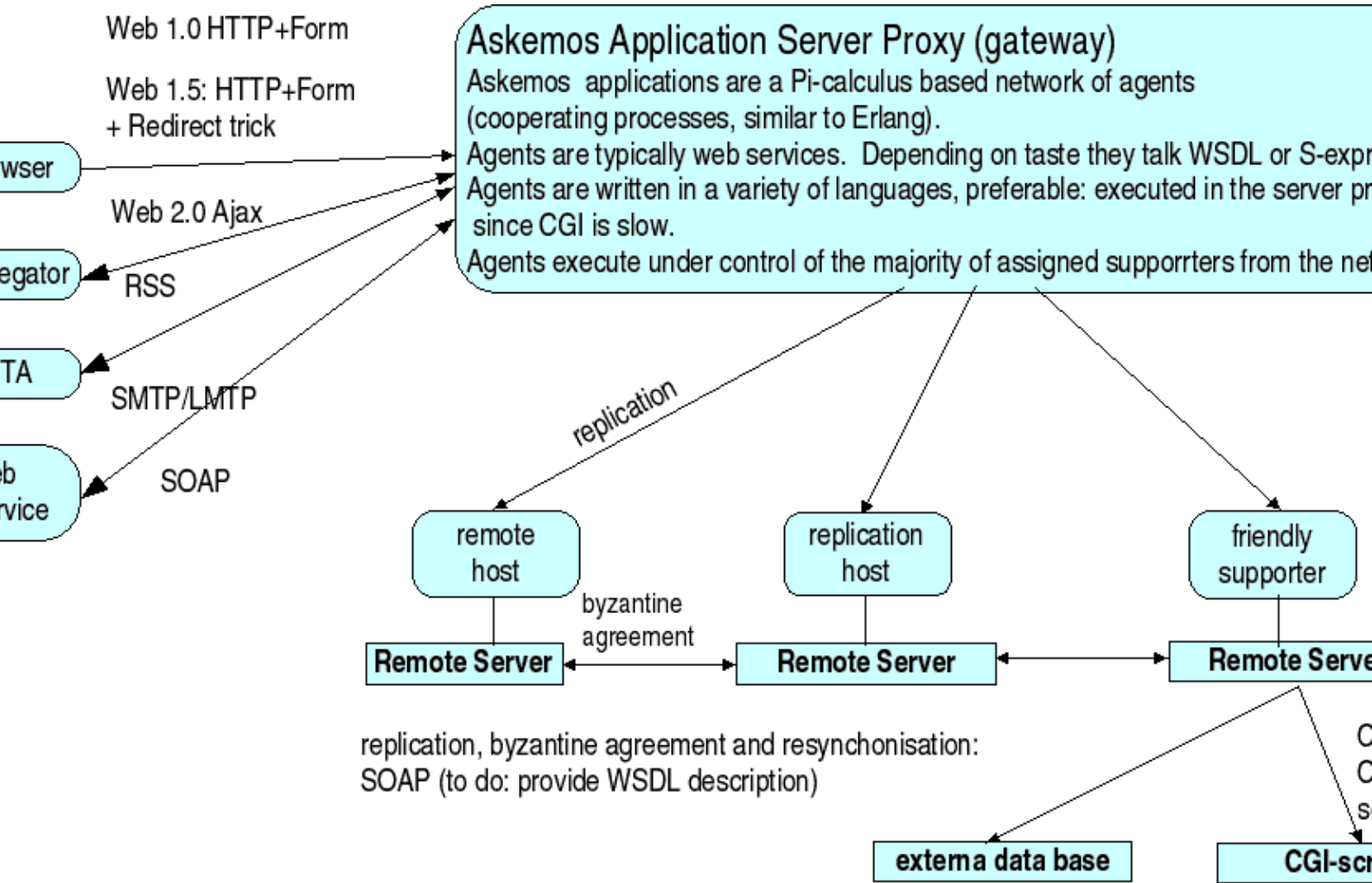
|             |                     |
|-------------|---------------------|
| Purpose     | Operating System    |
| Manufacture | softeyes            |
| Platform    | <u>POSIX / BSD</u>  |
| License     | <u>GPL</u>          |
| Market      | distributed systems |

## About

The software "askemos" is an incorruptible and intrusion resistant agent oriented operating system. It forms the basic infrastructure (TCB) required for the project Askemos. The AskemosDesign defines an autonomous, virtual machine on document level, which works synchronous among distributed, independant components (companies, departments etc.).

## Big picture

Usually it's not easy to grasp what Askemos is, since it has a few facets. It's importance lies in the combination and seamless integration of a certain set of features found elsewhere too. Nowadays a comparison with couchdb might be helpful.



This [white paper](#) used to be the best one about the design. While it's ageing, it's still a recommended read.

Askemos is expected to be implemented on top of various base systems. (As it's best deployed with one instance at the local host, the next logical step would be to integrate it into the browser. But we are not yet there.) This document describes implementation and administration of the Askemos system available today.

## Basics

The current implementation is a two layer operating system built on the foundation of a Unix compatible base system, which does the I/O. The AskemosServer compares to the "exec server" of micro kernel systems.

There are plenty of approaches already, but none was found, which could satisfy the AskemosDesign requirements. Askemos can fulfill those because it overcomes by design a lot of accidental hardness as inherent in legacy systems. Today's operating systems provide abstractions for individual computers, network connections files etc. these have no importance for the developer or user in the Askemos.

Askemos is an environment where simple, collaborating agents may form an intelligent systems. My personal dream of the latter is sort of a public library which can't "burn".

### 1. [quick overview](#)

2. a more [general description](#)
3. [Askemos Einführung](#) (german, different content)
4. [frequently asked questions](#)

## Development at top of Askemos

There are roughly two ways of dynamically generating programm output like websites. One way is the "PHP way" (or Perl, Ruby, pick one). This means you write some [HTML](#) code and mix the output from running another program in between. Beeing often used, there are several disadvantages though. For example, operation on fragments of code must all be done on the string-level. Superflous, distracting and potentially buggy code is introduced in your application to parse input and generate output. At worst, your output is simply invalid. Another disadvantage is that malicious or obnoxious [HTML](#) and scripts can be inserted relatively easy in the output by any potential attackers of your site, unless you take great care to escape [HTML](#) characters - which again clutters you application code.

The other way is to use [XML](#): You use languages like [XSL](#), which includes [XSLT](#) and [XPath](#) or [XQuery?](#). The [BALL kernel](#) blends the latter approach with [MIME converters](#) to use simple input syntax like [SXML](#), [Wiki](#) or [OOo](#), since reference [XML](#) is needlessly verbose and thus hard to write, [Scheme](#) and [SQL?](#) into a system of [communicating sequential processes](#) to express your business logic.

- [example tour](#) introducing of [XSLT](#) methods for the [AskemosDVM](#)
- Function documentation: [scheme](#), [dsssl](#), [xpath](#), [srfi's](#), [others \(index\)](#)
- [core api](#) for interaction with the Askemos kernel.
- [AskemosProtection](#) a capability system without central authority.
- Applications on top of Askemos bear similarities with the software behind the yahoo store. Get some background from <http://www.paulgraham.com/lib/paulgraham/bbn excerpts.txt>
- Tip: use [Ajax / Web2.0](#), but don't depend on it (client side scripting might be turned off in sensitive environments).

## [BALL](#) internals

Most references to background information are in source code comments, where used.

- How to write new virtual machines. (Example: `policy/nu.scn`)
- source repository [ModuleStructure05](#)

## Installation and Administration

- [system requirements](#)
- [installation](#)
- **Licence** The Askemos server is available under the [GPL](#).

If you feel the need to license under other conditions, please contact [softeyes.net](#)

- [performance considerations](#)

---

Last modifikation: Wed, 16 Jun 2010 16:44:57 +0200

Author(s): jfw,

Document number A849640f672ed0df0958abc0712110f3c page AskemosServer delivered to public at Wed, 08 Sep 2010 15:48:37 +0200