

# Semasys - Technical Reference

Albert Weichselbraun (albert@iaeste.at)  
Franz Xaver (fx@ung.at)

20th March 2003

## Abstract

Semasys is an *open, distributed* platform for internet services, featuring high availability and easy maintenance.

This document is meant to give an impression about the scope and usability of Semasys and how it can be used within your environment.

## 1 Introduction

Internet services are crucial for most companies operations and off-times could create considerable cost.

Therefore different approaches to improve the reliability of services as such have been taken. These efforts often focus on

- the hardware-level, e.g. redundant hard disks via RAID, power supply units, UPS
- the connectivity (multiple backbones)
- inner and outer security (off-shore or underground housing-centres).

Nevertheless it's obvious that all these approaches are futile because there is still *one* point of failure - even if we are able to decrease the probability that this *one* point may fail.

On the other side it has to be mentioned, that the more sensitive data is stored in such dedicated housing-centres the more that centres are a rewarding potential target to all kind of attacks (denial of service, hacking, etc. as well as physical attacks).

To overcome these difficulties Semasys has been developed, featuring a completely different approach.

Semasys gains high availability via redundant, distributed data-storage, where the term "distributed" means, that the data is stored in different, fully independent nodes, allocated in distinct geographic locations.

Therefore this system can provide the user with a new quality of availability - even

if one, two or even more nodes are down for several reasons - the system will stay completely functional.

Beside from the high availability aspect there is an additional benefit from the distributed character of the system, bringing the services closer to the customer - like commercial web-caching services [2].

One of the main benefits of Semasys is its *openness* - everybody, who wants to participate in the system can do so, by using a set of scripts, creating a Semasys-cell with an arbitrary number of partners.

## 2 Semasys - Requirements

Semasys provides a standardised interface for independent *nodes*, forming a distributed, highly available *Semasys-Cell*.

### 2.1 Hardware

All nodes have to full fill a set of requirements, to participate in a particular *Semasys-Cell*.

It's important to understand, that these requirements can vary from cell to cell, so that the cells efficiency is highly dependent on these criteria, covering:

- Connectivity,
- CPU,
- RAM and
- Hard-disk capacity.

### 2.2 Software

- a UNIX based operation system (Linux, \*BSD, AIX, Mac OS X)
- rsync, ssh, djbdns [3]
- Free UID's from 60000-65000

On each node a Semasys-directory-tree with the following contents will be created (Table 1).

## 3 Semasys - a technical view

The following sections present the key-concepts included into the Semasys architecture.

<code>/semasys</code>	Semasys Root Directory
<code>/semasys/bin</code>	Semasys Binaries and Scripts
<code>/semasys/nodes/{nodename}</code>	Semasys Nodes-Home
<code>/semasys/nodes/{nodename}/dns</code>	Semasys DNS-Zones
<code>/semasys/public</code>	Common redundant Semasys Data
<code>/semasys/private</code>	Per node Semasys Data

Table 1: Semasys Filesystem-Structure

### 3.1 Common “Distributed” Filesystem

In this context distributed doesn’t mean, that the filesystem is synchronised in real-time.

Instead it is divided into a common-public-part, redundant available at *all* nodes and a private part only present on one node and meant to be used for modifications and testing before its contents are released to all nodes.

Users therefore will write their data (e.g. webpages) into the `/semasys/private` directory and after they finished their changes the data is committed and distributed via `rsync` to the appropriate public-directory on the other (and the users own) nodes.

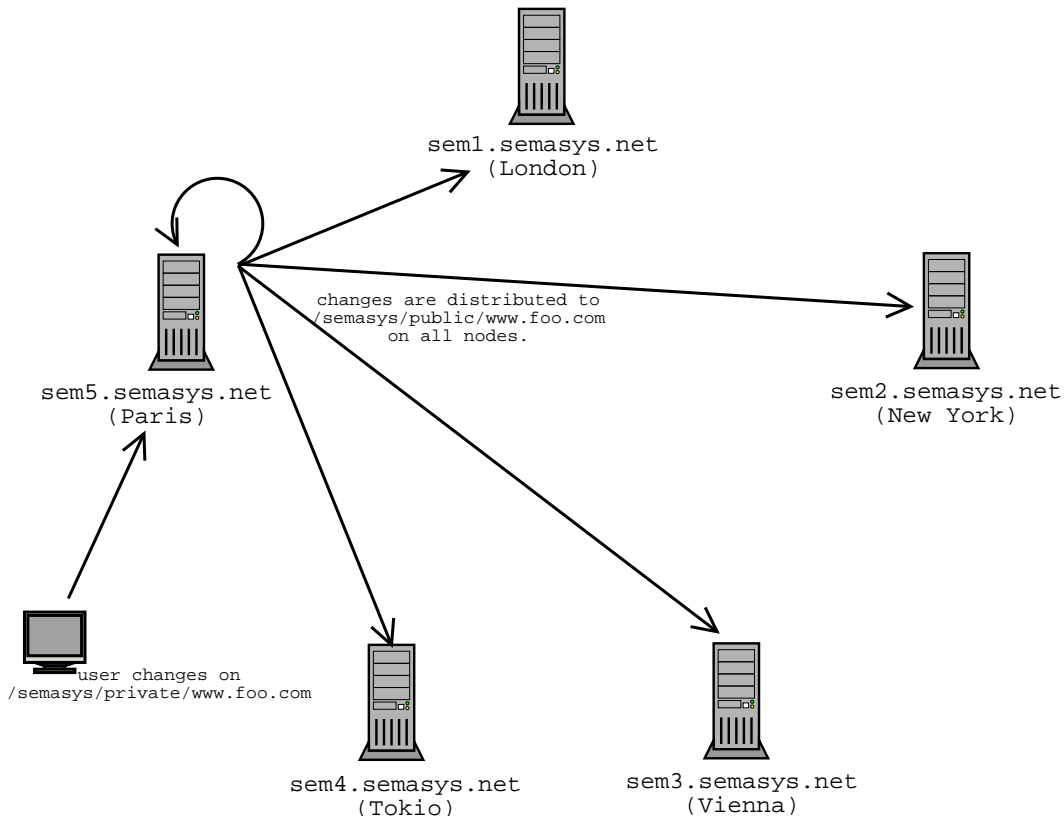


Figure 1: Filesystem distribution

### 3.1.1 User ID's

Due to the distributed structure of Semasys, the assignment of user-ids has to be regulated in some way. Therefore depending on the hosts configuration the UID-range from 1000-60000 might be used for local-users, while 60001-65000 is reserved for semasys-host-UIDs.

When data is committed, it is distributed to *all* nodes, using the Semasys-host-UID of the node containing the `/semasys/private` directory for this data. Figure 2 shows the UID-usage on a Semasys node. The example node uses the UIDs starting from 1000<sup>1</sup> to host the local users private-filesystem.

In the public directory a replica of the hosts own data (with UID 60001) as well as the data which is maintained by the other hosts (with their respective UIDs) is stored.



Figure 2: Filesystem

Another important point to note is, that every node is *responsible* for the distribution of the data of *it's* private-directories via `rsync`. Therefore every host has:

1. a private ssh-key, authorising it against the other nodes.
2. the other hosts home-directories (`/semasys/nodes`) with the respective public ssh-key in `/semasys/nodes/nodename/.authorized-keys`.

## 3.2 Common Distributed DNS

High available services are useless, if the underlying DNS-servers are down, so that customers cannot reach the (still working) hosts.

Therefore an approach for distributing DNS has been included in Semasys, whereby the zone-data is located in `/semasys/nodes/{nodename}/dns` and updates via

---

<sup>1</sup>it is not required that a node starts with UID 1000 for the first user - it's the node-administrators own business, how to distribute the UID's in the reserved range - as long, as the UIDs starting from 60000 are free for the other hosts public filesystems

rsync.

All zone-files are polled in regular intervals by each node and the zone-data is copied to the nodes DNS-master-database, which is served using Bernsteins's djbdns [3], excluding non-reachable hosts.

Due to the fact that the *non-reachable* status of a server is determined on a per node basis, this configuration is even stable in the case of a net-segmentation, so that all services covered by Semasys are still fully functional.

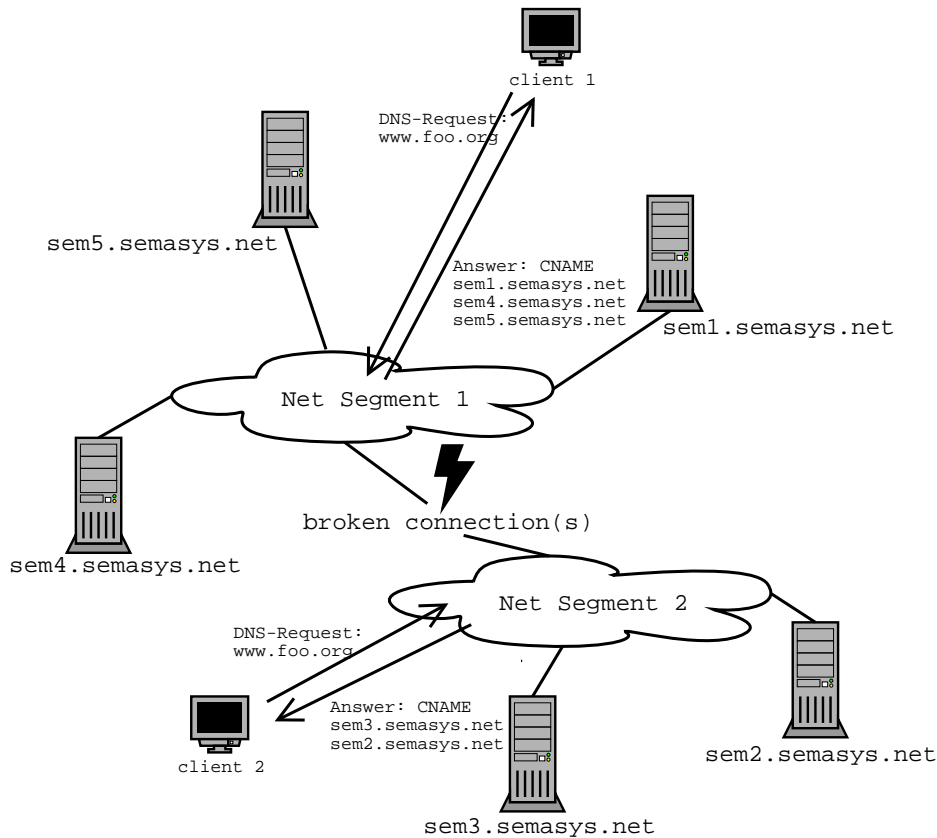


Figure 3: DNS-Answers in case of a net-segmentation

### 3.3 Common Distributed Databases

## References

- [1] High-Availability Linux Project  
<http://www.linux-ha.org/>
- [2] Akamai  
<http://www.akamai.com>
- [3] djbdns  
<http://cr.yip.to/djbdns.html>