Project no.	034084
Project acronym:	SELFMAN
Project title:	*Self Management for Large-Scale Distributed Systems based on Structured Overlay Networks and Components*

European Sixth Framework Programme Priority 2, Information Society Technologies

Periodic Activity Report - Year Four (M37-M40)

Due date of deliverable:	Nov. 15, 2009
Actual submission date:	Nov. 15, 2009
Start date of project:	June 1, 2006
Duration:	40 months
Dissemination level:	CO

Contents

1 Executive summary

The SELFMAN project aims at building large-scale distributed systems that are self managing. SELFMAN has two main results: a scalable self-managing transactional storage (Scalaris and Beernet) and a scalable real-time media streaming application (PeerTV). Other results include the Kompics reactive component model for distributed systems, the SicSim discrete event simulator, the MyP2PWorld network and concurrency emulator, and the Distribution Subsystem of the Mozart Programming System version 1.4.0.

In the final four-month project extension, we have continued this work in six directions. We have build a simulation and emulation environment for Kompics. We have improved performance of the Scalaris storage service by optimizing the underlying protocol. We have made an initial protocol, called Salute, for dynamic slicing of resources in an overlay network. We have made a collaborative drawing application, called DeTransDraw, which uses Beernet on gPhones running the Android operating system. We have made an in-depth evaluation of Beernet, and we have studied self-protection mechanisms for spam resistance which can be applied to Wikipedia and other systems with anonymous content.

In this final period, the SELFMAN project received significant media attention. Two articles were published in ICT Results. These articles were referenced by ACM Technews and taken over almost verbatim by ScienceDaily and many other online technical magazines and blogs. A third article appeared in the RobotArmageddon blog making a serious comparison of SELFMAN to Skynet (from the Terminator films). Finally, a fourth article appeared in *Automatisering Gids* ("*Automation Guide*") in the Netherlands.

2 Project objectives and major achievements

The project extension continued the project for four months, from M37 to M40. Its aim was to continue the momentum of SELFMAN: to provide additional results, to complete existing results, and to make a bridge toward future projects. We have made the following additional deliverables in the extension:

- D2.4: Simulation and emulation environment for Kompics P2P framework (partner KTH). This deliverable is part of task T2.2 and extends D2.1c.
- D3.4: Optimizations for self-managing global storage services (partner ZIB). This deliverable is part of task T3.2 and extends D3.2b.
- D4.5: Third report on self-configuration support (partner INRIA). This deliverable is part of task T4.1 and extends D4.1b.
- D5.9: Distributed mobile application on gPhone (partner UCL). This deliverable is part of task T5.8 and extends D5.8.
- D5.10: Design and analysis of Beernet, the Mozart structured overlay network implementation (partner UCL). This deliverable is part of task T5.3 and consists of the Ph.D. dissertation of Boris Mejias.
- D5.11: Self-protection mechanisms which provide spam resistance (partner NUS). This deliverable is part of task T5.6 and extends D5.6.
- D6.1d: Second project workshop. The workshop was held on Sep. 15, 2009 in conjunction with SASO 2009.

In addition to these deliverables, there were also many additional publications which completed work that had been ongoing in SELFMAN. These are given in Section 6. Finally, there has been significant dissemination work, including the final project workshop, articles in the technical press, advertisements, collaborations, new project proposals based on SELFMAN, and invited talks. These are given in Section 5.

Milestones, deliverables, and person-month status Table 1 gives the final milestones for the project at M36. Table 2 gives the deliverables for the period M25-M40 (the last 16 months). Table 3 gives the person-month utilisation for the same period M25-M40.

Milest. no.	Milestone name	WP no.	Date due	Actual delivery date	Lead contractor
M4.2	Finished self-* services on architectural framework	WP4	M36	M36	INRIA
M5.3	Finished demonstrator applications.	WP5	M36	M36	UCL, ZIB, KTH
M5.4	Understand effectiveness of self-* services for the application.	WP5	M36	M36	FT & others
M5.5	Understand how to build self managing applications.	WP5	M36	M36	UCL
M6.3	Understand general principles of self management.	WP6	M36	M36	UCL & others
M6.4	Understand effectiveness of two platform implementation.	WP6	M36	M36	UCL & others

Table 1: Milestones list for period M25-M40

Del. no.	Deliverable name	WP no.	Date due	Actual date	Est. indic.	Used indic.	Lead contr.
D2.4	Simul. and emul. env. for Kompics P2P framework	WP2	M40	M40	6	0	KTH
D3.1c	Final rep. on formal models for trans. over SON	WP3	M36	M36	2	2	ZIB
D3.2b	Rep. on repl. stor. serv. on SON	WP3	M36	M36	6	3	ZIB
D3.3b	Rep. on simple DB query layer for repl. stor. serv.	WP3	M36	M36	3	3	ZIB
D3.4	Opt. for self-man. global storage services	WP3	M40	M40	5	3	ZIB
D4.1b	2nd rep. on self-config.	WP4	M36	M36	2	6.5	INRIA
D4.1c	Self-config. support (SW)	WP4	M36	M36	12	18	INRIA
D4.2b	2nd rep. on self-healing	WP4	M36	M36	3	3	KTH
D4.2c	Self-healing support (SW)	WP4	M36	M36	8	8	KTH
D4.3b	2nd rep. on self-tuning	WP4	M36	M36	5	5	ZIB
D4.3c	Self-tuning support (SW)	WP4	M36	M36	11	11	ZIB
D4.4b	Self-protection support	WP4	M36	M36	10	10	NUS
D4.5	3rd rep. on self-config.	WP4	M40	M40	2	4	INRIA
D5.2b	Demo. app. for J2EE (SW)	WP5	M36	M36	6	6	KTH
D5.3	Demo. app. for Mozart (SW)	WP5	M36	M36	4	4	UCL
D5.4a	Qualitative eval. of autonomic features of apps.	WP5	M36	M36	7	5	FT
D5.4b	Quantitative eval. of autonomic features of apps.	WP5	M36	M36	8	6	FT
D5.6	Eval. of security mech.	WP5	M36	M36	2	2	NUS
D5.7	Guide for self-man. apps.	WP5	M36	M36	8	6	UCL
D5.8	Distributed mobile application on gPhone	WP5	M36	M36	2	2	UCL
D5.9	2nd rep. on dist. mobile application on gPhone	WP5	M40	M40	1	1	UCL
D5.10	Design & analysis of Beernet	WP5	M40	M40	1	1	UCL
D5.11	2nd rep. on eval. of security mech.	WP5	M40	M40	1	1	NUS
D6.5c	Final progress and assessment report with lessons learned	WP6	M36	M36	4	4	UCL

Table 2: Deliverables list for period M25-M40

3 Workpackage progress of the period

During the period M37-M40, the work was concentrated on finishing the extra deliverables of the project extension. There were no significant deviations from this plan.

3.1 Workpackage 1: Structured overlay network and basic mechanisms

Workpackage 1 is completed; no work was done on it during M37-M40.

3.2 Workpackage 2: Service architecture and component model

Objectives Complete the deliverable D2.4.

Progress KTH has publicly released a stable implementation of the SELFMAN architecture and component framework in the Kompics open source project (`kompics.sics.se`). The framework offers basic primitives for self-configuration and self-healing. A Peer-to-Peer component framework was implemented together with tools for evaluating P2P systems. KTH released a simulation and emulation environment for P2P systems built with Kompics. Real system implementations (in Kompics) can now be executed both in reproducible simulation and real-time emulation, locally, or in distributed deployment.

Deviations and corrective actions No significant deviations.

Deliverables Deliverable D2.4: Simulation and emulation environment for Kompics P2P framework.

3.3 Workpackage 3: Self-managing storage and transactions

Objectives Complete the deliverable D3.4.

Progress KTH and ZIB have worked on designing a component architecture for the transactional DHT algorithms, using the Kompics component model (D3.1c). KTH contributed to optimizing the Scalaris system where we reduced the validation phase of the commit protocol from 6 to 4 communication steps. A Kompics implementation of the transactional key-value store is currently underway. KTH has been working on an algorithm to maintain consistent views on node responsibilities in a DHT, among replication groups. This is currently being leveraged for building a DHT that offers an atomic consistency model. Several conference submissions are underway.

Deviations and corrective actions No significant deviations.

Deliverables Deliverable D3.4: Optimizations for self-managing global storage services.

3.4 Workpackage 4: Self-management services

Objectives Complete the deliverable D4.5.

Progress Basic mechanisms for self-configuration (D4.1b) and self-healing (D4.2b) were implemented in the Kompics public release. The work on self-healing of overlay networks (in the face of network partitions) has been completed by designing efficient algorithms for the merging of partitioned overlay networks. An important component in managing peer-to-peer networks is to handle NAT traversal. KTH together with Peerialism classified all types of NATs and specified the major methods for the discovery of NAT properties and their traversal. KTH has designed a preliminary methodology for using different types of feedback loops in self-managing of distributed systems which was illustrated in a self-managing distributed file-system.

INRIA investigated the problem of deploying and configuring applications in a large scale WAN environment in presence of churn. In particular, we have completed an initial study of a framework and protocol, called Salute, for the dynamic slicing of resources in an overlay network, which includes a notion of node profile to take into account the resource requirements of applications and the volatility of nodes. To set up an infrastructure for dynamic slicing of resources in a real-world WAN environment, we have investigated the question of building a NAT-resilient gossip-based peer-sampling service.

Deviations and corrective actions No significant deviations.

Deliverables Deliverable D4.5: Third report on self-configuration support.

3.5 Workpackage 5: Application requirements and evaluations

Objectives Complete the deliverables D5.9, D5.10, D5.11.

Progress UCL worked on the DeTransDraw application on gPhones. This turned out to be much harder than expected, primarily due to the many gaps in the Android system. This is documented in deliverable D5.9, which supersedes the previous deliverable D5.8. UCL also continued work on the evaluation of Beernet, which has led to a mostly complete version of the Ph.D. dissertation of Boris Mejias. This work has also led to a new release of Beernet.

For D5.11, NUS worked on extending the Wiki self-protection infrastructure of D5.6 to help identify spam. We deal with the common case where spam comes from virtually created identities or anonymous users. We also report on a result in routing in small world networks that extends D5.6. With a small modification, the routability of the Watts and Strogatz model drastically increases and can approach the results in Kleinberg networks. This work suggests that the navigability of specially constructed small world networks will not match the structure of actual social networks such as Facebook.

Deviations and corrective actions No significant deviations.

Deliverables

- Deliverable D5.9: Distributed mobile application on gPhone.
- Deliverable D5.10: Design and analysis of Beernet.
- Deliverable D5.11: Self-protection mechanisms which provide spam resistance.

3.6 Workpackage 6: Management, dissemination, and exploitation

Objectives Complete the project management and the project workshop.

Progress The second SELFMAN workshop was held at SASO 2009 (Sep. 15, 2009) (see Section 5.1). In addition to project management activities, various dissemination activities were held including advertisements and interviews. The latter led to several articles in the technical press.

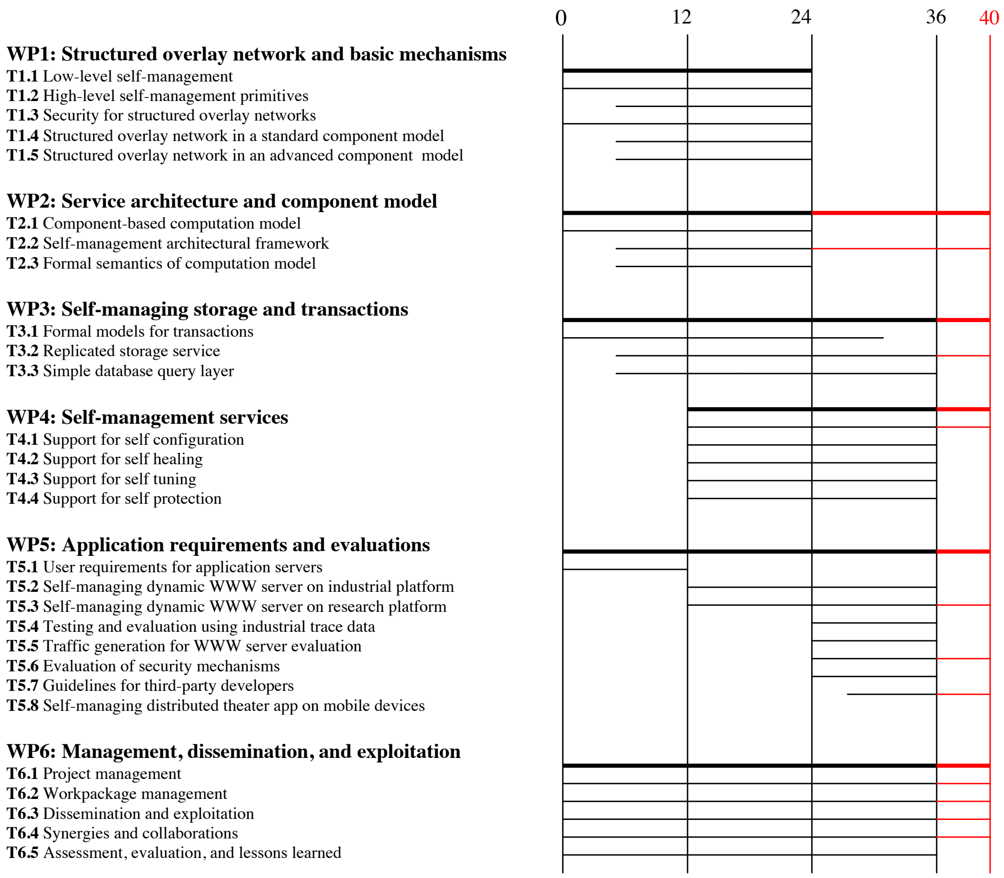

Figure 1: Project timetable at end of project (M40)

Deviations and corrective actions No significant deviations.

Deliverables Deliverable D6.1d: Second project workshop.

4 Consortium management

Figure 1 gives the project timetable at the end of the project. It shows the four-month extension and the tasks that were extended. The extension is part of the amended Description of Work dated April 10, 2009. Table 3 gives a detailed breakdown of the person-months spent by each partner in the final period from M25 to M40 (last year plus four-month extension). Note that ePlus remains in the table, although with zeroes, and Peerialism is the eighth partner.

Person-Month Status Table															
		TOT	UCL	KTH	INR.	FT	ZIB	EP	NUS	PR	**ACT**	UCL	KTH	ZIB	NUS
WP1 Structured overlay network and basic mechanisms	**Act tot**:	**0**	0	0	0	0	0	0	0	0	**0**	0	0	0	0
	Plan tot:	**51**	14	10	8	0	0	0	17	2					
WP2 Service architecture and component model	**Act tot**:	**11**	3	0	8	0	0	0	0	0	**0**	0	0	0	0
	Plan tot:	**76**	17	18	19	11	11	0	0	0					
WP3 Self-managing storage and transactions	**Act tot**:	**11**	0	0	0	0	11	0	0	0	**4**	4	0	0	0
	Plan tot:	**48**	6	11	0	8	20	0	0	3					
WP4 Self-management services	**Act tot**:	**65.5**	0	27	17.5	0	8	0	9	4	**6.9**	4	1.9	0	1
	Plan tot:	**76**	6	14	19	3	15	0	17	2					
WP5 Application requirements and evaluations	**Act tot**:	**34**	0	0	0	5	9	0	16	4	**11**	10	0	0	1
	Plan tot:	**47**	9	5	6	9	11	0	3	4					
WP6 Management, dissemination, and exploitation	**Act tot**:	**12.1**	0	0	2.6	3	3	0	3.5	0	**5**	4	0	0	1.5
	Plan tot:	**30**	10	3	3	5	3	0	3	3					
Total project person-month	**Act tot**:	**133.6**	3	27	28.1	8	31	0	28.5	8	**27.4**	22	1.9	0	3.5
	Plan tot:	**328**	62	61	55	36	60	0	40	14					

Estimation of AC own personal cost		
AC institution	Cost/month	Total cost for period
UCL	3*6250/month	18750
	15*4600/month	69000
	4*4000/month	16000
KTH	1.9*10500/month	19950
NUS	2*1522/month	3544
	1.5*6500/month	9750

Table 3: Person-month status table (contract no. 34084, acronym SELFMAN, period 1/6/2008 to 30/9/2009)

4.1 Peerialism

As reported in the Year 3 activity report, GGF (Global Gaming Factory X) tried to acquire the file sharing site The Pirate Bay (the eighth most popular site on the Internet) as well as Peerialism. Peerialism would have become the technology provider for a legal version of The Pirate Bay, at an acquisition price of 10 MEuro. The deal did not go through due to financial problems at GGF.

The acquisition attempt has provided a global visibility for Peerialism and its unique technologies, which were developed partly in SELFMAN. Peerialism is ranked among the 33 hottest tech startups in Sweden in 2009 according to the Swedish technology newspaper NyTeknik [1] and is one of the top three technology companies in Europe according to the Seedcamp community [2]. At the time of writing, Peerialism is being praised as a company that is promoting the use of research-based P2P technology in innovative ways for media distribution.

4.2 Coordination activities

- Boris Mejías, Jérémie Melchior, and Yves Jaradin. "Decentralized transactional collaborative drawing". Demonstrator in Internet of Service 2009. Collaboration meeting for FP6 & FP7 projects, Brussels, Belgium, Jun. 10-11, 2009.

4.3 Partner visits

- Seif Haridi made several visits to ZIB for cooperation on Scalaris and related research.
- Peter Van Roy visited KTH in Aug. 2009 to work on the CACM invited paper.

5 Plan for using and disseminating the knowledge

5.1 Second project workshop: Architectures and Languages for Self-Managing Distributed Systems

We held the Workshop on Architectures and Languages for Self-Managing Distributed Systems in conjunction with SASO 2009, Sep. 14-18, 2009, San Francisco, CA.

Workshop goal The goal of the workshop is to bring together researchers and practitioners interested in the construction of self-managing distributed systems. It will place the emphasis on software engineering (especially, software architecture and component-based software engineering) aspects of this construction, including models, architectures, languages, control techniques, middleware and tools that can be used to support the modular and principled building of self-* distributed systems. Topics of interest for the workshop include (but are not limited to):

- Component models and architectures for self-management.
- Generative and reflective (including aspect-oriented) techniques for self-management.
- Languages for self-managing systems, including formal specification, architecture description, programming, and domain specific languages.
- Control techniques for self-managing systems, including control-theoretic and decision-theoretic techniques.
- Coordination and decentralized architectures of control.
- Analysis and verification techniques for self-managing systems.
- Middleware and tool support for self-managing distributed systems.
- Algorithms for distributed self-management, including event detection, distributed control, etc.

Application areas of interest to the workshop include (but are not limited to): Web services, social networks, cloud computing, P2P systems and applications, pervasive computing.

Organizing committee

- Jean-Bernard Stefani, INRIA, Grenoble, France
- Seif Haridi, SICS & KTH, Stockholm, Sweden
- Peter Van Roy, Université catholique de Louvain, Belgium

Program committee

- Gordon Blair, Lancaster University, UK
- Pierre Cointe, Ecole des Mines, Nantes, France
- Thierry Coupaye, Orange Labs, France
- Jean-Charles Fabre, Institut National Polytechnique, Toulouse, France
- Seif Haridi, SICS & KTH, Stockholm, Sweden
- Tom Holvoet, Katholieke Universiteit Leuven, Belgium
- Mark Jelasity, University of Szeged, Hungary
- Emre Kiciman, Microsoft Research, Redmond, WA, USA
- Mark S. Miller, Google Research, USA
- Alexander Reinefeld, Zuse Institute, Berlin, Germany
- Bradley Schmerl, Carnegie-Mellon University, Pittsburgh, PA, USA
- Jean-Bernard Stefani, INRIA, Grenoble, France
- Alexander Wolf, Imperial College, London, UK
- Peter Van Roy, Université catholique de Louvain, Belgium

Invited talk Terence Kelly, HP Labs, "Discrete control theory for self-managing systems".

Accepted papers

- Thorsten Schütt, Alexander Reinefeld, Florian Schintke, and Christian Hennig, Zuse Institute Berlin, Germany. "Self-Adaptation in Large-Scale Systems: A Study on Structured Overlays Across Multiple Datacenters".
- Mikael Högqvist and Stefan Plantikow, Zuse Institute Berlin, Germany. "Towards Explicit Data Placement in Scalable Key / Value Stores".
- Boris Mejias and Peter Van Roy, Université catholique de Louvain, Belgium. "From mini-clouds to Cloud Computing".

- Artur Andrzejak, Zuse Institute Berlin, Germany. “Generic Self-Healing via Rejuvenation: Challenges, Status Quo, and Solutions”.
- Cosmin Arad, Tallat M. Shafaat, and Seif Haridi, Royal Institute of Technology, Stockholm, Sweden. “Self-distributing Software Updates through Epidemic Dissemination”.
- Felix Halim, Yongzheng Wu, Roland H.C. Yap, National University of Singapore. “Routing in the Watts and Strogatz Small World Networks Revisited”.
- Pedro Javier del Cid, Nelson Matthys, Danny Hughes, Sam Michiels, and Wouter Joosen, Katholieke Universiteit Leuven, Belgium. “Resource management middleware to support self managing wireless sensor networks”.
- Lin Bao, Ahmad Al-Shishtawy, and Vladimir Vlassov, Royal Institute of Technology, Stockholm, Sweden. “Policy Based Self-Management in Distributed Environments”.

5.2 Articles in the technical press

As far as we know, four different articles appeared in the technical press:

- Two articles on SELFMAN appeared in ICT Results on Sep. 30, 2009 and Oct. 2, 2009. These articles were based on a one-hour telephone interview between journalist Robert Adler and Peter Van Roy, complemented by information from the SELFMAN website. The ICT Results articles were widely reproduced in online technical newsmagazines and technical blogs, including ACM Technews (Oct. 5, 2009), ScienceDaily, MegaPlatinum, VWN News, NUZE.ME, Newstin, Fast Company, ReadWriteWeb, PhysOrg.com, Technology.am, AlphaGalileo, and The Web Scene. Translations appeared in French, Russian, and Portuguese. We reproduce the two original articles in Sections 5.2.1 and 5.2.2.
- A third article appeared in the RobotArmageddon technical blog on Oct. 3, 2009. This article seems to be inspired by the ICT Results articles. The article makes a radical extrapolation from the SELFMAN goals: it compares SELFMAN to Skynet (c.f., the Terminator films, where the artificial intelligence Skynet takes over the world). The article is certainly correct in that more advanced versions of SELFMAN

with artificial intelligence algorithms will eventually be developed. Section 5.2.3 gives an extract from that article.

- A fourth article appeared in the Dutch technical newspaper *Automatisering Gids* ("*Automation Guide*") on Oct. 16, 2009 (number 42) (in Dutch). This article is based on the ICT Results articles and a series of email exchanges between journalist Thijs Doorenbosch and Peter Van Roy. The article appears on page 11 in the newspaper and is referenced on the cover page. We give an English translation of the article in Section 5.2.4.

5.2.1 The self-managing, "unbreakable" internet? (ICT Results)

High-powered internet applications typically need teams of experts to maintain them. Not any more, say European researchers who have built a system to create applications that manage and fix themselves.

Part of the internet's potential lies in its ability to link hundreds, thousands, or even millions of devices. Whether a user is downloading a video from a peer-to-peer service, performing scientific research on a grid, or using "cloud computing" to manage a business, programs that let many devices and applications work together are crucial. The problem, says Peter Van Roy, coordinator of the EU-supported SELFMAN project, is that it's getting harder to keep those systems working. "The central challenge when you build big internet applications is how to keep them running without having to tweak and manage them all the time," he says. The SELFMAN team set out three years ago to solve that problem by finding out how to build programs that take care of themselves in the rough-and-tumble internet environment. "We wanted to make big internet applications easy," Van Roy says, "so that all the management problems you normally have are handled by the system itself." The payoff, he says, will be huge. "It will take the internet to the next level."

Self-management – four key features The SELFMAN researchers identified four vital functions for a distributed application to manage itself – self-configuring, -tuning, -healing and -protecting.

Software is continually being patched, updated or replaced. For a distributed system to configure itself, it needs to keep track of all its components, update them as needed, and make sure that all parts of the system can still talk to each other. "Our system can ask a component, what version are you? Who are you talking to? It can then replace an old version with a new one as needed," says Van Roy.

Self-tuning means that the system can instantly adjust to changing loads and to components leaving or joining the network. “Suppose one node is getting overloaded,” says Van Roy. “Our load-balancing algorithm allocates new nodes close to that hotspot. It spreads the heat to the other nodes and the hotspot cools down.”

The internet is an unpredictable environment. Routers crash, cables get cut, parts of the system overload and grind to a stop, and components come and go. “With SELFMAN,” Van Roy says, “each node stores some of the data and each piece of data is replicated a certain number of times. If a node crashes, the other nodes detect the crash, find a new node and give it the missing data. The system heals itself.”

One of the biggest problems SELFMAN tackled was self defence. The researchers discovered that a system’s security depends on its topology – how nodes are linked to each other. They found that “small world” networks – in which most nodes are not directly linked, but in which any node can communicate with another in a few steps – were the safest. “With a small world network, it’s easier to detect, isolate, and eject bad nodes,” says Van Roy. “The security service observes the system’s behaviour. If it notices that certain parts of the network are acting abnormally, it takes action.”

It’s all in the architecture The SELFMAN team found that building these advanced capabilities into useful applications required a highly structured approach. The foundation of each application is a structured overlay network. That’s a program – itself replicated across the network – that keeps track of all the nodes and connections between them, and can decide when and how to fix problems. The next level is a replicated storage system. It makes sure that each node has access to the same data, and that data are always replicated to ensure they do not disappear.

The third level houses SELFMANs transactional problem-solver. It relies on a sophisticated algorithm called Paxos to provide a systematic way of reaching consensus among any number of fallible components. Van Roy uses the analogy of a transfer between two bank accounts. “If you want to reduce one bank account by 100 euros and add that 100 to another, you want both or nothing,” he says. “Each node must see the same data.” “Getting all this fluid behaviour – where even if nodes are crashing or new nodes are coming in or the network has problems it never blocks the system – was a big technical problem,” says Van Ray [sic]. “We needed Paxos to get it to work.”

The SELFMAN architecture and components have been used to build some impressive applications. These include a prize-winning distributed Wikipedia that can handle far more queries than the current version, a commercially successful media streaming service, and a graphics program that lets multiple users collaborate on a design.

Van Roy believes that SELFMAN opens the door to a host of high-powered, flexible, and "unbreakable" internet applications. "Right now we're just scratching the surface," he says.

Several SELFMAN-inspired applications will be highlighted in a second ICT Results story The SELFMAN project received funding from the ICT strand of the EUs Sixth Framework Programme for research.

5.2.2 Self-managing internet applications flex their muscles (ICT Results)

A European research project that incubates self-managing internet applications is paying off. It has inspired a Wikipedia thats better than the original and super-efficient streaming video, with more to come.

The European research project SELFMAN has created a programming system that makes it much easier to build high-powered distributed internet applications that manage themselves. Distributed applications tap the internets power by enabling hundreds, thousands, or even millions of devices and programs to link up and work together. These are the invisible programs that let users bank, buy and play games online, search for information, share files, and network with applications like Facebook and Twitter. SELFMAN ensures that all those moving parts stay in touch, up-to-date, and work together smoothly without the teams of experts that have been needed to tweak, patch or protect the system as components come online or drop out, or in response to communication or component breakdowns, and even deliberate attacks.

"We want to make this kind of application easy, so people can really take advantage of this huge network of computers and small devices," says Peter Van Roy, SELFMAN's coordinator based at the Université Catholique de Louvain, Louvain-La-Neuve in Belgium. SELFMAN's programming architecture and set of components have already sparked new applications that utilise and showcase its capabilities. "It's the first time that we've been able to get large internet applications with many computers talking together, when all the applications can take care of themselves," says Van Roy. "Now we are taking this to the next level."

What's wilier than Wikipedia? When you pitch a question to Wikipedia, you're likely to be one of some 2000 people trying to access it that second. Wikipedia manages those 2000 requests per second, plus the constant stream of updates to its database of articles, by using at least 16 powerful database servers arranged in a master-slave architecture. Researchers at the Zuse Institute Berlin (ZIB), one of SELFMAN's academic partners, decided to test the SELFMAN approach by implementing Wikipedia's core database in a flexible, scalable and self-managing distributed system. They based their Wikipedia on Scalaris, an open-source distributed database system that incorporates SELFMANs self-configuring, self-tuning and self-healing capabilities.

"Every piece of information in our Wikipedia exists four times – that's a number that we found works well – and is spread over all the nodes," says Van Roy. "A clever algorithm called Paxos does the heavy lifting to make sure the nodes all agree." They expected that their Wikipedia would have several advantages, including higher reliability, lower operating costs, and scalability – the ability to add new capacity easily in response to increasing demand. The Scalaris-based Wikipedia met all those expectations. "With our wiki, if ten times as many people start to use it, you just add more machines," says Van Roy. "We can handle far more users than the actual Wikipedia." In addition, since Scalaris takes care of itself in the unpredictable internet environment, the costs of keeping it running are significantly reduced. ZIBs Scalaris-based Wikipedia won first prize in the Institute of Electrical and Electronic Engineers 2008 Scalable Computing Challenge. "It really shows the kind of things we want to do," says Van Roy.

Streaming video on demand The wide applicability of SELFMAN's toolkit is evident in a completely different application – PeerTV, produced by another SELFMAN partner, Stockholm-based Peerialism.

"They had a very different problem," says Van Roy. "PeerTV needed to manage load between flowing streams of video data coming from a few big servers but going to millions of clients." They solved the problem of streaming all that video reliably and efficiently by optimising peer-to-peer (P2P) networks with SELFMAN's ability to balance loads and transfer data efficiently.

"Until now, P2P wasted enormous amounts of bandwidth because of its random routing of data," says Van Roy. "Some service providers had to block certain P2P applications." In contrast, PeerTV connects components through a well-defined, constantly optimised topology. That 'topologically aware' routing reduced network traffic by two-thirds. Overall, PeerTV found that they could provide the same quality of service as established video providers, easily scalable to an unlimited number of viewers, but with a cost reduction of from 50 to 90 percent. PeerTV has a direct commercial interest. Peerialism developed it for the Swedish company MPS Broadband AB, and Peerialism itself is currently being acquired by GGF.

Van Roy is excited by the varied applications that are starting to flow from SELFMAN. These include small applications, such as a program that lets users work together on a graphic design using their gPhones, and massive ones, such as a load-balancing system for France Telecom. But he is even more excited by SELFMAN's implications. "The actual smartness of the internet comes from applications like Google or Wikipedia, not from the plumbing that just shuffles data around," says Van Roy. "SELFMAN takes those smart components and amplifies them – makes them automatically more efficient, more scalable and more robust."

This is the second of a two-part special feature on SELFMAN The SELFMAN project received funding from the ICT strand of the EUs Sixth Framework Programme for research.

5.2.3 Skynet Part II: Will these projects lead to a new Skynet? (RobotArmageddon)

Internet based applications will soon be able to defend and heal themselves

A project is currently being developed that is attempting to create internet based applications that configure, manage, heal and defend themselves. It is called SELFMAN, and it could be the future of internet apps. Cloud computing and non-disc based apps are all the rage right now, but SELFMAN takes those concepts much, much further.

SELFMAN is able to configure and manage itself automatically by tracking down all of it's pieces, no matter where they are located, updating them on the fly and verifying communication between everything. By ensuring strong communication between its parts, SELFMAN is able to redistribute loads automatically to ensure constant uptime. This is also part of it's healing abilities. Normally, if part of a distributed application is overloaded, it may crash or disconnect, greatly affecting the usability and functionality of the program. SELFMAN coordinator Peter Van Roy understood the need to make the program self sustaining.

> "With SELFMAN," Van Roy says, "each node stores some of the data and each piece of data is replicated a certain number of times. If a node crashes, the other nodes detect the crash, find a new node and give it the missing data. The system heals itself."

SELFMAN is also able defend itself from outside attacks. To accomplish this, communication is not linked from one part of the program to all of the other parts, but rather each part can communicate independently to whichever part it needs. This makes the program's network smaller overall, so if one portion of the program comes under attack, it can be blocked by the rest of the program without the attack spreading directly across the predefined lines of communication like a virus.

Projects like SELFMAN are frightening because more advanced versions with artificial intelligence will eventually become available. How can I be so sure about this? Since the program is self-sustaining, the most efficient way to handle it would be through artificial intelligence. An intelligent program working inside the internet that can configure, defend and heal itself could pose *major* problems, depending on the level of intelligence the program contained and the purpose of the program.

5.2.4 Smart webapplication repairs itself (Automatisering Gids)

Web 2.0: Algorithm takes the barb out of managing distributed processes

- *Management no longer needs a team of experts.*
- *Distributed applications are more scalable.*

Distributing applications over the Internet offers enormous computing power. For compute-intensive tasks but also for applications that are used by many people at once, distributed systems offer many advantages.

From the management viewpoint, the lack of control over these users is a problem. Version control and security are tough when applications are shared over many thousands or maybe even millions of physical devices. Furthermore, the Internet is a dynamic network: network nodes can appear or disappear without any notification. This means that for reliable business applications this form of distributed computing power can seem very unattractive. That can change when applications become smart enough to manage themselves. The Université catholique de Louvain took up the gauntlet and built a 'self-healing' platform for distributed internet applications together with partners in the SELFMAN project.

Project coordinator Peter Van Roy: "The big difference with cloud computing is that the servers in a cloud architecture are in one (virtual) datacenter connected by a fast and reliable network. Control is centralized on one computer, with a redundant implementation, and the whole acts like an enormous server. The starting point in SELFMAN is exactly the opposite. All computers are distributed over the Internet, connected through a possibly unreliable network and there is no single manager."

Instead of a single manager, the participating computers (called computing nodes or nodes) manage each other. The foundation of the system is the same as the technology that underlies peer-to-peer networks. Each computing node in the network can perform the same tasks. When one of the nodes crashes, another takes over its tasks. To avoid losing data, each node keeps a copy of part of the data that is being processed in other nodes. A consensus algorithm ensures that the different nodes frequently exchange data about the software version, the other nodes that are being contacted, and the tasks that are being executed. When needed, the software version is updated or the next batch of work is provided. A load-balancing algorithm makes sure that none of the nodes is overloaded. When needed, the algorithm can add or remove nodes and distribute the missing data.

According to Van Roy, this system is amazingly fast. In a comparison with the Wikipedia datacenter, the SELFMAN-built Scalaris library achieves similar performance with the same number of servers, except that Scalaris can improve performance simply by adding servers. Van Roy: "When computers crash, Scalaris does not stop but at worst just slows down."

System security was one of the hardest problems. The researchers observed that the creation of many small groups of connected nodes worked the best. In this organization, it is quickly clear when 'bad' nodes are introduced. The algorithm isolates these nodes and switches them out of the system.

The SELFMAN project developed three applications as 'proof of concept' One of them, PeerTV, is further being developed by the Swedish service provider Peerialism for the distribution of videos over Internet. It offers its clients the possibility of showing live videos to large numbers of visitors at low cost. Peerialism claims that the system is friendly for Internet service providers because as much traffic as possible is handled locally.

PeerTV uses a combined server- and peer-to-peer approach. The video information is on a server, but the end users' computers help reduce the load on the servers. "Video streams are divided into 'stripes', where each stripe only needs a fraction of the bandwidth." The algorithm takes into account the bandwidth to each user and the relative distance to other users in the network.

The Zuse Institute Berlin, one of the SELFMAN partners, used the Scalaris platform to reconstruct the heart of Wikipedia. For this the institute was awarded the first prize in the 2008 International Scalable Computing Challenge from the IEEE standards organization. Van Roy: "The system works best for Internet applications where people collaborate and the system has to maintain coherence. We have also developed a decentralized drawing tool where people work together on the same drawing. The same approach can be effective for many other Web-2.0 applications such as social networks and YouTube."

Whether all web applications are suited for Scalaris, Van Roy cannot say. "Probably not, but that is still an open question. We have created the basic infrastructure in three years, but we did not have the time to investigate the best use of the system." On the other hand, he expects that many applications can be made SELFMAN-ready. "It is actually rather easy for applications that use a database. The database can be replaced by Scalaris or a similar system. Right now, this still needs a custom interface because the Scalaris API is not the same as that of MySQL. Possibly there will be a standard API in the future."

The SELFMAN project was formally completed last week. Scalaris and a derived system Beernet are available under an open-source license. Van Roy says he is already busy submitting successor projects. "We want to exploit the possibilities of these storage systems as much as possible."

5.3 Advertisements

In the final period, we made a threefold advertisement with PSCA:

- We have placed a final Web banner for SELFMAN on the PSCA European Union website (for one year, from Sep. 2009 to Sep. 2010). This banner will link to the SELFMAN project website.
- We will have a one-page article in Public Service Review which will appear early 2010. This article will present the final results of SELFMAN.
- We will have a one-page editorial in the Public Service Review will will appear early 2010. The editorial is an opportunity for us to present our vision on the current state and future evolution on peer-to-peer and cloud computing. It is more general than just SELFMAN.

5.4 Collaboration and use

- Scalaris was released under the Apache Open Source License. This implementation is under continuing development by a community of developers, not just from within the SELFMAN project but also including outside collaborators. The code is available at `scalaris.googlecode.com`.

- Scalaris is being evaluated and used by several companies. We know of the following companies who have been evaluating Scalaris for their own use:
  - Linden Lab, maker of the Second Life virtual world application.
  - StudiVZ, a large German social network for college and university students in Europe considered to be a rival to Facebook.
  - Plurk, a social network considered to be a rival to Twitter.
  - searchmetrics.com, a search optimization company.
  - immobilienscout24.de, a German real estate agent.
- Beernet and its relaxed ring structure are used in the following work in collaboration with Peter Van Roy's research group at UCL:
  - DeTransDraw and its gPhone implementation were realized in collaboration with the Belgian Laboratory of Computer-Human Interaction (BCHI), headed by Jean Vanderdonckt at UCL.
  - The RELEASeD research group at UCL, headed by Kim Mens, is implementing the relaxed ring in Objective-C to use in Apple devices including the iPhone.
  - The SOFT laboratory at the Vrije Universiteit Brussel is planning to use the relaxed ring and the feedback methodology for context-aware systems, in collaboration with UCL.

5.5 Project proposals

The following project proposals are successors to SELFMAN and were submitted during the final period:

- PHASEMANIA: An FP7 STREP proposal in FET Open on using reversible phase transitions to build extremely robust self-managing Internet applications. UCL, ZIB, SICS, and Peerialism are partners. The peer-to-peer techniques developed in SELFMAN, in particular the transaction and merge algorithms used in Scalaris and Beernet, will be key starting points.
- CLOUDMAN: An FP7 STREP proposal on a decentralized self-managing cloud infrastructure. UCL, ZIB, SICS, Peerialism, and INRIA are partners. The Scalaris/Beernet technology developed in SELFMAN will be used as a key part of the peer-to-peer cloud infrastructure.

- VISION: An FP7 IP proposal on a federated data intensive storage cloud. KTH is a partner in the area of computational storage. The Kompics technology developed in SELFMAN will be used and further developed.

- 3C (Conjuring Clouds with Constraints): An Expression of Interest in the European ComplexityNet initiative. UCL is a partner in the area of self-managing distributed applications. Peer-to-peer techniques from SELFMAN will be used to implement the optimized self-managing cloud solution.

5.6 Invited talks and similar activities

- Boris Mejías, Jérémie Melchior, and Yves Jaradin. "Decentralized transactional collaborative drawing". Demonstrator in Internet of Service 2009. Collaboration meeting for FP6 & FP7 projects, Brussels, Belgium, Jun. 10-11, 2009.

- Boris Mejías. "Beernet: a relaxed-ring approach for peer-to-peer networks with transactional replicated DHT". Presentation given at the Doctoral Symposium in the "XtreemOS Summer School", Wadham College of the University of Oxford, Oxford, UK, Sep. 10, 2009 (Best Presentation Award).

- Cosmin Arad, Jim Dowling, and Seif Haridi. "Building and evaluating P2P systems using the Kompics component framework". Demonstrator and poster, 9th International Conference on Peer-to-Peer Computing (P2P 2009), Seattle, WA, Sep. 8-11, 2009.

6 Publications and software

6.1 Software releases

- Beernet: the relaxed pbeer-to-pbeer network, version 0.2. Programming Languages and Distributed Computing Research Group, UCL, Belgium. See `beernet.info.ucl.ac.be`. Sep. 30, 2009. There will be more releases after the end of the project.

- Kompics reactive component model for distributed computing, version 0.4.2.4. KTH, Sweden. See `kompics.sics.se`. Sep. 2009. There will be more releases after the end of the project.

6.2 Award

Boris Mejías won the Best Presentation Award of the six presentations given at the Doctoral Symposium in the XtreemOS Summer School:

- Boris Mejías. "Beernet: A Relaxed-Ring Approach for Peer-to-Peer Networks with Transactional Replicated DHT". Doctoral Symposium, XtreemOS Summer School. University of Oxford, UK, Sep. 10, 2009.

6.3 Dissertations and licentiate theses

- Alexandre Bultot. "A Survey of Systems with Multiple Interacting Feedback Loops and Their Application to Programming". Master's Thesis, Université catholique de Louvain, Belgium, Aug. 2009.
- Nico Kruber. "DHT Load Balancing with Estimated Global Information". Master's Thesis, Humboldt-Universität zu Berlin, Germany, Sep. 2009.

6.4 Conference papers

- Michael Lienhardt, Claudio Antares Mezzina, Alan Schmitt, and Jean-Bernard Stefani. "Typing Component-Based Communication Systems". Joint 11th IFIP WG 6.1 International Conference on Formal Methods for Open Object-Based Distributed Systems and 29th IFIP WG 6.1 International Conference on Formal Techniques for Networked and Distributed Systems (FMOODS / FORTE 2009). Springer Lecture Notes in Computer Science 5522, Jun. 9-12, 2009, pages 167-181.
- Cosmin Arad, Jim Dowling, and Seif Haridi. "Developing, Simulating, and Deploying Peer-to-Peer Systems using the Kompics Component Model". Fourth International Conference on COMmunication System software and middlewaRE (COMSWARE 09), Dublin, Ireland, Jun. 16-19, 2009.
- Anne-Marie Kermarrec, Alessio Pace, Vivien Quéma, and Valerio Schiavoni. "NAT-resilient Gossip Peer Sampling". 29th International Conference on Distributed Computing Systems (ICDCS 2009), Montreal, Québec, Jun. 22-26, 2009, pages 360-367.
- Jérémie Melchior, Donatien Grolaux, Jean Vanderdonckt, and Peter Van Roy. "A Toolkit for Peer-to-Peer Distributed User Interfaces:

Concepts, Implementation, and Applications". ACM SIGCHI Symposium on Engineering Interactive Computer Systems (EICS 2009), Pittsburgh, PA, Jul. 14-17, 2009.

- Roberto Roverso, Sameh El-Ansary, and Seif Haridi. "NATCracker: NAT Combinations Matter". 18th International Conference on Computer Communications and Networks (ICCCN 2009), San Francisco, CA, Aug. 3-6, 2009.
- Ahmad Al-Shishtawy, Vladimir Vlassov, Per Brand, and Seif Haridi. "A Design Methodology for Self-Management in Distributed Environments". 2009 IEEE International Symposium on Scientific and Engineering Computing (SEC-09), Vancouver, Canada, Aug. 29-31, 2009.
- Thorsten Schütt. "Gossip-based Topology Inference for Efficient Overlay Mapping on Data Centers". Short paper, 9th International Conference on Peer-to-Peer Computing (P2P 2009), Seattle, WA, Sep. 8-11, 2009.
- Xavier Etchevers and Thierry Coupaye. "Benchmarking Autonomic Systems from a Technical and an Economical Perspective". Third International ICST Conference on Autonomic Computing and Communication Systems (Autonomics 2009), Limassol, Cyprus, Sep. 9-11, 2009.
- Felix Halim, Yongzheng Wu, and Roland H.C. Yap. "Wiki Credibility Enhancement". Fifth International Symposium on Wikis and Open Collaboration (WikiSym 2009), Orlando, FL, Oct. 25-27, 2009.

6.5 Paper submissions and dissertation drafts

- Peter Van Roy, Seif Haridi, and Alexander Reinefeld. "Software design with interacting feedback structures and its application to large-scale distributed systems". Invited submission (Oct. 2009), CACM. Currently under evaluation.
- Boris Mejías. "Beernet: A Relaxed-Ring for Self-Managing Decentralized Systems with Transactional Replicated Storage". Ph.D. Dissertation draft, Université catholique de Louvain, Sep. 2009. (This is deliverable D5.10.)

6.6 Workshop papers and technical reports

- Boris Mejías, Alfredo Cádiz, and Peter Van Roy. "Beernet: RMI-Free Peer-to-Peer Networks". Proceedings of the First International Workshop on Distributed Objects for the 21st Century (DO21), colocated with ECOOP'09, Genoa, Italy, Jul. 7, 2009.
- Florian Schintke, Alexander Reinefeld, Seif Haridi, and Thorsten Schütt. "Enhanced Paxos Commit for Transactions on DHTs". Technical Report ZR-09-28, Zuse Institute Berlin. Sep. 2009.
- Boris Mejías and Peter Van Roy. "From Mini-Clouds to Cloud Computing". Workshop on Architectures and Languages for Self-Managing Distributed Systems, SASO 2009, San Francisco, CA, Sep. 15, 2009.
- Thorsten Schütt. "Self-Adaptation in Large-Scale Systems: A Study on Structured Overlays Across Multiple Datacenters". Workshop on Architectures and Languages for Self-Managing Distributed Systems, SASO 2009, San Francisco, CA, Sep. 15, 2009.
- Mikael Högqvist. "Towards Explicit Data Placement in Scalable Key / Value Stores". Workshop on Architectures and Languages for Self-Managing Distributed Systems, SASO 2009, San Francisco, CA, Sep. 15, 2009.
- Artur Andrzejak. "Generic Self-Healing via Rejuvenation: Challenges, Status Quo, and Solutions". Workshop on Architectures and Languages for Self-Managing Distributed Systems, SASO 2009, San Francisco, CA, Sep. 15, 2009.
- Cosmin Arad, Tallat M. Shafaat, and Seif Haridi. "Self-distributing Software Updates through Epidemic Dissemination". Workshop on Architectures and Languages for Self-Managing Distributed Systems, SASO 2009, San Francisco, CA, Sep. 15, 2009.
- Felix Halim, Yongzheng Wu, and Roland H.C. Yap. "Routing in the Watts and Strogatz Small World Networks Revisited". Workshop on Architectures and Languages for Self-Managing Distributed Systems, SASO 2009, San Francisco, CA, Sep. 15, 2009.
- Mikael Högqvist and Nico Kruber. "Active/Passive Load Balancing with Informed Node Placement in DHTs". Proceedings of the IFIP Fourth International Workshop on Self-Organizing Systems (IWSOS 2009). Springer. Zürich, Switzerland, Dec. 9-11, 2009.

References

[1] Here are Sweden's hottest technology startups (in Swedish), November 2009. Available at www.nyteknik.se/nyheter/innovation/tillvaxtforetag/article543431.ece.

[2] Mini Seedcamp Helsingborg winners, May 2009. Available at blog.seedcamp.com/2009/05/mini-seedcamp-helsingborg-winners.html.