Software Technologies
The Missing Key Enabling Technology

Toward a Strategic Agenda for Software Technologies in Europe

ISTAG
July 2012
Software Technologies

The Missing Key Enabling Technology

Toward a Strategic Agenda for Software Technologies in Europe

ISTAG – Information Society Technologies Advisory Group

Working Group on Software Technologies

(Draft July 2012)
# Table of Contents

The Missing KET 4

Toward a Strategic Agenda for Software Technologies in Europe 4

ISTAG Report on 2020 Key Software Technologies 4

Executive Summary 4

1. **Introduction** 6
   
   1.1 The KETs 7
   
   1.2 The missing KET 7
   
   1.3 This report 8

2. **Science of Programming and Software Engineering: Key Challenges** 10
   
   2.1 Software Engineering 10
   
   2.2 Software in 2020 and Beyond 11
   
   2.3 Towards a Roadmap 12

3. **Key Software Technologies for Parallel Computing, Cloud Computing and High Performance Computing** 15
   
   3.1 Parallel Computing 15
   
   3.2 Cloud Computing 16
   
   3.3 High Performance Computing 17

4. **Key Software Technologies for Building Data Value** 20

5. **Key Software Technologies for Social Computing** 22

6. **Key Software Technologies for Internet based applications and real time services** 24

7. **Key Software Technologies for Embedded Systems** 27

8. **Key Software Technologies for Human-Centred Computing and Multimedia** 30
   
   8.1 Human-Centred Computing 30
   
   8.2 Multimedia 31

9. **Key Software Technologies for Enterprise Application** 33
10. Key Software Technologies for Ensembles – the future generation of software intensive systems 36

11. Conclusion and recommendations 39
   11.1 Main recommendation 39
   11.2 Software Engineering 39
   11.3 Parallel, Cloud and High Performance Computing 40
   11.4 Building Data Value 41
   11.5 Social Computing 42
   11.6 Internet based applications and real time services 43
   11.7 Embedded Systems 43
   11.8 Human-Centred Computing and Multimedia 44
   11.9 Enterprise software 44
   11.10 Future generation of software intensive systems 45
Software Technologies: The Missing Key Enabling Technology

Toward a Strategic Agenda for Software Technologies in Europe

ISTAG Report on 2020 Key Software Technologies

Executive Summary

The objective of this report is to create sense of urgency in the European software industry and awareness about software as the prime industrial differentiator and basis for innovation. It also aims to raise awareness that missing the strategic importance of software technology as a key enabling technology will lead to a significant drawback for global competitiveness. This is particularly true for embedded system software, telecommunication software and enterprise software among others where Europe has built up four decades of worldwide leadership which is now endangered given that other regions have understood the relevance of this technology and started to catch up. The old approaches to software development are simply not sufficient for maintaining global leadership. To remain a strong competitor, the European software intensive industry must increase its efficiency and productivity, enhance software quality and reduce lead times.

We propose the creation of a European Software Agenda which could incorporate a long term research agenda coupled with an innovation agenda and the creation of new business models:

Main Recommendation: A Strategic Agenda for Software Technologies in Europe should be created in cooperation with Industry, Academia and Public sector. The agenda should outline the strength from a European perspective and how we can renew and strengthen it. Common goals and needs in the short, medium and long term should be described. It should also describe what actions should be implemented to achieve these goals and needs and how we can, in a smarter way, make use of existing efforts, resources and facilities. The agenda should also draw up proposals on how the strategic work should be organized, run and quality assured. The first version of this Agenda should be delivered before mid-2013.

This recommendation is supported by an analysis of current strengths and emerging trends in software technology and a number of critical application areas including: parallel, cloud and high performance computing; big data; social computing; internet-based applications and real-time services; embedded systems; human-centred computing and multi-media; enterprise applications and the future generation of software-intensive systems. Some of these areas are ones where
Europe has had a significant lead, others are areas where changing business and innovation models offer new opportunities for Europe.

The report includes a number of detailed recommendations which support the main recommendation and suggest actions that could be taken to support the implementation. These are collected together in the concluding section of the report. Key recommendations include:

- Europe should encourage the emergence of open source software repositories associated with development or qualification tools to gather and foster the result of cooperative R&D or local initiatives.
- Launch a European initiative on software approaches for advanced computing systems.
- Create a European Data Observatory that builds upon the open data initiatives for the public sector in Europe and extends it to the private sector. This would provide a universally accessible test bed.
- Develop interdisciplinary funding programmes to enable us to understand the concepts of social computing/computations, its societal value and the innovation and entrepreneurship possibilities that arise as a result.
- Support the effort that by 2020, software intensive real time systems should be executable on shared hardware and easily connectable to the outside world.
- Europe should develop new scientific foundations, system design methodologies, development processes and tools to create the technical solutions tackling the challenges posed by system complexity in the embedded systems domain.
- Develop a European strategic initiative on enterprise software technology to maintain Europe’s leadership.
- Set up a FET Flagship to support the right timescales, levels of ambition and long-term funding that would allow Europe to maintain its pre-eminent position in future generation software-intensive systems.

Software is a key driver for the European economy. It is important that we take action now to ensure that Europe remains at the forefront of this strategic technology.
1. Introduction

Arguably, one of the defining characteristics of the 20th Century was the information revolution: the systematic manipulation of information to improve all aspects of our daily lives. Just as the 19th Century industrial revolutions were powered by steam, the information revolution has been powered by software technology - without the software, computers would still be the tools of a high priesthood.

Software is a key driver for the European economy. Whilst the European software industry lags behind its international competitors, recent trends in technology (for example Future Internet, Mobile Computing and Cloud Computing) offer a window of opportunity to rectify this situation. The European software sector employs more than 2.75 million people and creates added value of €180B\(^1\). The European software market is the second biggest in the world with a 32% share of the global market in 2009; it was valued at €228.7B in 2008 and is projected to grow to €383.5B by 2020 – this rate of growth is twice the projected growth of the overall economy in the EU27 region.

The commoditisation of software and infrastructure inherent in the move towards Cloud Computing and the continued advances in Open Source and mobile applications should help to foster greater innovation. Europe must ensure that the younger generations are equipped with the appropriate technical and business skills to exploit these opportunities.

Non-functional features such as reliability and security will become more important differentiators. Increasingly software will be automatically produced and Europe needs to ensure that the appropriate production methods are employed to ensure that the European software sector is market leading in these aspects.

This report does not attempt to formulate specific recommendations to improve the competitive position of the European software industry. This would require the analysis of the outlook of the software market and the assessment of the current and future status of the European software industry, aims which are outside the scope and the timeframe of the report. However, it is evident the importance for Europe to remain at the forefront of this strategic technology, as software is, among other things, a large source of qualified employment.

The European software industry has a significant position only in few segments of the software field\(^2\) (examples are the areas of Embedded Systems and Enterprise Software). It lags behind its US competitors in important segments, like the traditional packaged software one. As a consequence, US vendors dominate the world Top 20 software vendor list.

Current technological developments (including the Internet of Services, the Internet of Things, the Mobile Internet, Cloud Computing and other new software delivery platforms) might change the dynamics of the world software industry. In this context, there is a window of opportunity for Europe to improve its competitive position in software.

---

\(^1\) Figures in this introductory section are taken from “Economic and Social Impact of Software and Software-based Services”, Smart 2009/0041, Pierre Audoin Consultants, August 2010.

\(^2\) For market trends and assessment of the European competitive position see “Economic and Social Impact of Software and Software-based Services”, Smart 2009/0041, Pierre Audoin Consultants, August 2010
1.1 The KETs

Key Enabling Technologies (KETs) are attracting strong interest at both Member State and European Union level, as they are seen as the route to new products, processes and services capable of generating economic growth and employment, and contributing to strengthening and/or rejuvenating existing European sectors. KETs will enable sustainable, smart and inclusive growth in Europe. In addition, KETs' follow-through applications will create the substantial jobs, growth and wealth required in our future European economies to remain competitive at global level. At the same time, they will make an essential contribution to the development of products and solutions to address grand societal challenges.

The European Commission selected, in its 2009 Communication, six KETs for Europe: nanotechnology, micro-nanoelectronics, advanced materials, photonics, industrial biotechnology and advanced manufacturing systems. The Commission based this selection on a screening of the common high-tech areas and strategies at Member State level. The selection criteria included their economic potential, their value adding and enabling role as well as their technology and capital intensity regarding R&D and initial investment costs.

A High Level Expert Group (HLG) on KETs was created with the task of elaborating a coherent European strategy to develop these six KETs. The HLG identified the major difficulties Europe has in translating its ideas into marketable products – in crossing the internationally recognised "valley of death". To cross this valley, it recommended a strategy comprising three pillars:

- A pillar focused on technological research
- A product demonstration pillar focused on product development
- A production pillar focused on world-class, advanced manufacturing.

By focusing on these key stages of the innovation chain, the HLG proposals trigger a virtuous cycle, from knowledge generation to market flow with feedback from the market to knowledge generation support, thereby strengthening economic development in Europe. Based on the three pillar bridge model, the HLG made a series of specific policy recommendations for a more effective industrial development and deployment of KETs.

1.2 The missing KET

As we daily use sophisticated office tools, it is easy to think that all of the software challenges have been solved. However, emerging technologies such as the Semantic Web, Internet of Things, Cyber Physical Systems, and Multi- and Many-core computers are bringing new fundamental challenges. It is unclear that we have the right programming abstractions or software tools to exploit these new developments to their full potential. Added to this, the software technologies to support the Internet of Services are still in their infancy and the tools and abstraction required to produce secure programs is an extremely active area of research. One can strongly argue that the “Future Internet”

---

software will be THE critical infrastructure on which all other critical infrastructures will depend, smart grids are just the first indicators for this. In other words, corrupted or dysfunctional software can disrupt complete societies significantly. Thus, software must be developed to be more robust and resilient. This is far from any App-like solution that one could easily download onto a mobile device.

To give their full power software technologies have to be embedded in an object or a device, providing intelligence to the system. It is largely admitted that all the devices will sooner or later embed software capabilities. Hence the six KETs will be useless if they are not strongly coupled with more or less specific software technologies.

Symmetrically, for creating, designing and manufacturing a new device, software technologies are unavoidable: it is well known that before existing in reality, a device exists virtually or digitally. Not only has the device, but also its environment, may exist digitally allowing a complete simulation of the behaviour of the object. This includes in particular a digital simulation of the way humans interact with the device.

Software technologies are also key in social interactions, not only for simulating those interacting but also for enabling new interaction modes. Web technologies and social networking are one of the most spectacular technological advances in the evolution of society. This soft (r)evolution ends the information era and opens a new era: the digital society.

1.3 This report

The objective of this report is to create a sense of urgency in the European software industry and awareness that software has become the prime industrial differentiator and basis for innovation and to raise awareness that missing the strategic importance of software technology as a key enabling technology will lead to a significant drawback in global competitiveness. This is in particular true for embedded system software, telecommunication software and enterprise software among others where Europe has built up four decades of worldwide leadership which is now endangered given that other regions have understood the relevance of this technology and started to catch up. The old approaches to software development are simply not sufficient for maintaining global leadership. To remain a strong competitor, the European software intensive industry must increase its efficiency and productivity, enhance the software quality and reduce lead times. We propose the creation of a European Software Agenda which could incorporate a long term research agenda coupled with an innovation agenda and the creation of new business models.

Therefore, in the framework of Horizon 2020, it would be essential to implement actions supporting the creation in Europe of fresh industrial capabilities in software (e.g. building/improvement of application platforms, of expected service and product offerings, of industrial processes tied to the creation, delivery and usage of innovative software-based services for different customer segments), equivalent to an extent to the pilot manufacturing lines recommended in the KETs report.

At the same time it should be recognized that, in order to succeed, a policy supporting the European software industry requires the adoption of a wide range of actions, far beyond the funding of
research and innovation activities. In the next section we will review some of the key challenges that need to be overcome in software engineering. The remaining sections focus on application areas and the challenges that they pose for software technologies. We draw our main recommendations together at the end of the report.

**Recommendation 1:** A Strategic Agenda for Software Technologies in Europe should be created in cooperation with Industry, Academia and Public sector. The agenda should outline the strength from a European perspective and how we can renew and strengthen it. Common goals and needs in the short, medium and long term should be described. It should also describe what actions should be implemented to achieve these goals and needs and how we can, in a smarter way, make use of existing efforts, resources and facilities. The agenda should also draw up proposals on how the strategic work should be organized, run and quality assured. The first version of this Agenda should be delivered before mid-2013.
2. Science of Programming and Software Engineering: Key Challenges

There is an on-going debate on the extent to which the writing of programs is an art, a craft or an engineering discipline. Viewing programming as an art concentrates on the creative aspects of programming; viewing it as a craft concentrates on the tools and methods but also emphasizes the bespoke nature of software; viewing programming as part of engineering discipline dictates that it is well-founded on an appropriate science base. In general, good programming is considered to be the measured application of all three, with the goal of producing an efficient and evolvable software solution (the criteria for "efficient" and "evolvable" vary considerably).

2.1 Software Engineering

Software engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software. It is the application of engineering to software because it integrates significant mathematics, computer science and practices whose origins are in engineering. It is also defined as a systematic approach to the analysis, design, assessment, implementation, test, maintenance, reengineering and quality of software, that is, the application of engineering to software.

In most branches of technology (and in commerce, administration, sales, management, medicine, ... as well), the valley of death between proof-of-principle and successful commercialisation is already being bridged. By software! It is software that propagates innovative ideas from a scientific research laboratory to their profitable application by working engineers throughout the world. It is software that simulates, explores, checks, optimises, integrates, automates, delivers and maintains the innovation in marketable product lines, under expert and specialised engineering direction. Software has become the prime industrial differentiator and basis for innovation. The end users hardly ever care about the intricacies of the software or how it is implemented. From the end users’ point of view the software simply has to work. This requires the software to be developed in a mature way, conformant to standards and best practices.

However in software itself the necessary software bridging methods, tools and theory to cross the valley of death are not available, or not universally deployed. Within individual large companies, the tool chains and management practices for software development have evolved to meet the commercial or technological needs of the individual companies. But there has not been the push (nor the pull) to attempt to co-ordinate and integrate a suite of Design Automation tools suitable for general purposes and covering the entire software life cycle. In recent years the development of extensible open-source tools and frameworks, such as Eclipse, NetBeans or IntelliJ IDEA for Java development, seems to offer a solution that allows companies to share the huge cost of developing and maintaining the core infrastructure while providing enough flexibility to tailor the tools to individual needs or provide commercial add-ons.
2.2 Software in 2020 and Beyond

By 2020, the old paradigm of software programs developed for a single hardware platform or operating system will be obsolete. Programs will be developed without basic knowledge of their runtime targets. Commodity computing systems will range from smart phones to the cloud, each of these platforms featuring hardware parallelism and hardware heterogeneity. The efficiency of applications on all platforms will highly depend on the ability of computing systems providers (from the smart phone manufacturer to the cloud manufacturer) to provide software technology that tailors applications to the underlying hardware at deployment and run time. The challenge will be to allow portability of both functionality and performance on the whole range of hardware platforms. The providers currently develop and simulate systems (software and hardware) in a mixed and integrated development environment. Another important aspect that needs improvement is the art of reaching solutions where hardware and software are as independent as possible. A better environment for understanding the consequences of different implementation alternatives would also be of great value in answering questions about what should be developed in software, in hardware or as a service. Whilst it is difficult to predict the precise operating environment for software in 2020 and beyond, it is to be expected that:

- An increasing reliance on software-driven artefacts will require that the software as infrastructure; i.e., software that is «always-on».
- Most software will run on distributed, heterogeneous and highly parallelised systems.
- Such systems will have to operate in environments which are open-ended and only partially observable.
- Requirements will change during the lifetime of the software requiring that it is evolvable and adaptable.
- Complexity will increase dramatically for each new project.
- Integration and testing of legacy software, open source and other third party software are issues in dire need of new levels of understanding.
- Software components will be combined in such a way that the resulting solution will provide new functionality emerging out of existing software components and services.
- Software quality will be increasingly important.

The required tool chain includes a target independent static compiler analysis and optimization platform, a dynamic run-time code optimisation infrastructure and a run-time workload deployment system to adapt the application to underlying hardware as well as tools supporting software auditing, simulation and testing. We might expect to see new programming or modelling languages which include adaptation mechanisms as «first-class citizens». Tools should have high level check functions – at the design level or system rather than the code level – and support instant feedback to the user, e.g. by simulation. Developing an understanding for cost efficient quality assurance should also be high on the agenda. The software should be developed for minimizing maintenance efforts, e.g. with possibility to upgrade continuously, in the field, sometimes decades after its initial deployment. Our understanding of software functionality must improve to a state where we have development environments supporting expressiveness, formalism and consistency to allow the software intensive systems to do exactly – and only – what it is designed to do.
In addition to this operational-level tool chain, novel software engineering management methods are urgently needed to reduce the still overly high rates of failure and cost overruns of large-scale software-intensive projects, especially but not only in the public sector:

- Novel methods will be required for the estimation of project cost and time in the dramatically more complex settings described above. This includes different kinds of risk assessment and methods for dealing with them: staffing issues in a globalized world in an aging Europe; frequent ex-post changes of plan due to competitive pressure or political context changes or new technological options; financing problems in a volatile financial setting; new business models for bringing software-intensive systems to the market.
- Requirements-level monitoring concepts must be extended by explicit consideration of risk aspects and early warning signals. It will be necessary that these concepts and related tools can be handled on the user side, as prosumer settings become increasingly important.
- Methods to stop unsuccessful projects gracefully, and more importantly, to recover failed but critical projects without excessive costs need to be linked to the technical quality and flexibility goals mentioned above.

2.3 Towards a Roadmap

The ever-increasing complexity and openness of software intensive systems forces us to move away from the 'intelligent design' approach, which currently rules software engineering. Instead, we must open the path for meta-design approaches as well as self-combining software systems and social computation. These approaches will acknowledge the inability of software engineers to foresee all possible situations that systems, connected to the open physical world, have to face. So, these approaches will focus on building software entities that have the ability to collaborate and spontaneously search for new design solutions in an autonomous and decentralized fashion. Natural phenomena and concepts from complex systems will be a major source of inspiration for unconventional models of architecture, communication and computation. We can expect to see a move towards continuous development instead of maintenance. It will be important that the new design solutions guarantee non-functional properties such as quality-of-service, security, privacy, etc... They should also support traceability of changes between code, requirements and reasons to ensure that improvements and adaptations are preserved through these changes. It is also inevitable that there will be an increasing need to be able to model stochastic and continuous phenomena in software systems.

Of course, the necessary integration of Software Engineering methods and tools must be based on a sound scientific theory, which must be strong enough to cover the major attributes of software today, for example, object orientation, non-determinacy, concurrency, distribution, timing, multicore. It must be strong enough to explain the success of large bodies of legacy software widely used today. There is not much current evidence of a consensus among computer scientists on the basis for such a theory. Furthermore, to cross the valley of death, the theory must be incorporated in industrial-quality methods and tools of general applicability. Such methods, tools and theory need to be tested and evolved in the further experimental investigation by scientists, before they will be fit for general application by software engineers. We need to ensure working interchange of software tools.
Software engineering goes far beyond coding and includes topics such as theoretical foundations for languages, models and systems, requirement handling and understanding, architecture design, system design, integration, specifications, documentation, verification, test, software quality, security-by-design and other higher-level activities. Creating brand new programs is a rare task and software engineering has to focus less on creating systems from scratch and instead emphasize techniques and tools for extending and changing existing systems. Furthermore, it is important to focus on empirical studies to compare and validate the effectiveness of different approaches to software engineering instead of relying on mostly unsubstantiated claims of superiority, as is the current practice. Universities need to focus more on software engineering as an academic discipline to develop and accredit graduates who possess the wide set of skills necessary to successfully build software: strong theoretical foundations paired with a willingness to grasp industrial demands on complexity, efficiency and quality, as well as social, linguistic and management abilities. It goes without saying that this will not happen unless industry, academia and relevant government bodies rise to this particular challenge with a new attitude.

**Recommendation 2.1:** The scale and duration of the development of existing and new methods, tools and theory is not well supported by the recent research culture, and even less by the funding culture. It will require collaboration of the researchers with the directors of funding policy to outline a strategy for culture change, leading particularly towards larger-scale and longer-term collaborations. The more mature branches of science have shown the way. The role of scientific competition against objective and realistic criteria must also be exploited.

**Recommendation 2.2:** A better environment for understanding the consequences of different implementation alternatives (e.g. quality, robustness, performance, maintenance, evolvability, ...) would also be of great value in answering questions about what should be developed in software, in hardware or as a service. We need to ensure working interchange of SW tools as well as methods and tools for dealing with risks and effective recovery from failure situations in software projects. Europe should set up an directory on good practices and exemplar educational programs for example on energy efficient software, code optimization, best in class software tools, large scale efficiency management,... Europe should foster methods and tools for improved software engineering project management, risk assessment, quality control, and problem recovery in a coherent manner between the technical, the user, and the management perspectives.
Recommendation 2.3: Europe should encourage the emergence of open source software repositories associated with development or qualification tools to gather and foster the result of cooperative R&D or local initiatives. Such repositories should not only give access to large databases of software components, solutions, protocols and open API, open data format but also help characterize the quality, security and licence information of these software items and foster the community creating or using them. These repositories could be based on current services like berlios.de or ow2.org.

Parallelism has become a mainstream technology in the development of present day computing systems impacting the whole spectrum of the computing landscape: from personal systems (smartphones, tablets, PC) to large-scale data centers and high-performance petascale computing systems. Parallelism is everywhere with an increasing degree as it is the only alternative to significantly augment the performance of computing systems while providing better autonomy for small devices or keeping the energy footprint at a lower level for data centers and HPC systems. This evolution has fundamental impacts on software technologies because parallelism has to be managed by software rather than hardware. The following paragraphs will highlight the key software technologies to exploit the quintessence of modern processors, Cloud systems and high-end HPC systems.

3.1 Parallel Computing

As the frequency race is over, caused by the energy wall, processor performance is mostly driven by an increasing number of cores, either homogeneous or heterogeneous, sometimes offering multithreading capabilities. To be able to exploit the performance of future processors, applications will have to exhibit a high degree of parallelism, typically 100,000 concurrent threads. Programming languages will have to provide mechanisms, through either parallel constructs or annotations, to express concurrency within applications hiding as much as possible the underlying hardware architecture. Another alternative is the definition of parallel programming skeletons for legacy applications. Both approaches will require dedicated runtime systems able to schedule threads across the available homogeneous or heterogeneous cores while managing data transfers and keeping the data coherent.

Developing the most appropriate architecture and applying new technology such as multi-core or parallel computing where it makes best sense will become prime competencies, but need more research for full understanding and for reaching a suitably ambitious goal of increased productivity. Raising the level of abstraction is essential for handling the variability issues with improved design efficiency. The solutions must be paired with possibilities to execute, verify and test the software on all levels, with tools giving fast feedback to the user, using simulation and other validation and verification technologies.

Resulting key software technologies:

- High-level parallel languages capable of handling heterogeneous many-core processors
- Parallel algorithmic skeletons that can be composed
- Unconventional, implicitly parallel, programming paradigms
- Runtime systems combining dynamic scheduling and memory management
3.2 Cloud Computing

The last five years have shown the emergence of Cloud Computing, implementing the vision of computing and information processing as a utility. It shares the same vision with Grid Computing that emerged in the end of the 90’s, which paved the way to this new computing paradigm. The Cloud Computing concept can be realized through several implementations, at different layers and exposed to the users as services. Infrastructure-as-a-service (IaaS) provides on demand a virtualized computing or networking infrastructure to users and concerns the management of large-scale data centers providing the basic infrastructure to develop Web applications and to process Big Data. Platform-as-a-service (PaaS) offers an access to an application development environment to develop applications based on the composition of a set of generic services. Software-as-a-service (SaaS) provides a complete application to users. One of the fundamental aspects of Cloud Computing is to provide elasticity and scalability. These two intrinsic properties impact not only the way resources are managed but also forces the application programmers to design their applications so that they can be run on a unknown set of resources that even fluctuates during the execution. Elasticity requires thus to exploit virtualization, through the use of virtual machines that can be migrated and to use specific parallel programming models, such as the Map and Reduce paradigm. Such programming models will have to interact in a most effective way with data management systems to process and analyze large, heterogeneous and dynamic data (see Key Software Technologies for Big Data). Cloud Computing raises however two concerns which have a strong societal impact: energy consumption and trust, privacy and security preservation. Proliferation of data centers will increase significantly the electricity consumption, however reduction of the energy footprint must be solved in a global way including the network infrastructures as well as mobile devices.

The trust, privacy and security concern is motivated by factors like the vast amount of personal data being stored and processed, the lack of an interoperable trust-enhancing identity management framework and the difficulty of providing guaranteed quality of service for security, privacy and dependability due to scale, heterogeneity and complexity, compounded by the inherent lack of accountability given the uncertainty on jurisdiction and law enforcement. The increased level of vulnerability affects both consumers and public and private organizations.

The relevant legal framework will need to be updated and technologies will have to be developed to comply with regulations and laws.

Resulting key software technologies:

- Virtualization to provide elasticity and scalability,
- Highly scalable and autonomic data management systems associated with programming models for processing Big Data,
- A service architecture to preserve privacy and security complying with European regulations, including assurance of security in storage and processing, interoperable encryption and key management, incident response and disaster recovery, physical security of the core network and the critical nodes,
- Smart energy management from data centers to mobile devices.
Given the expected economic impact of Cloud Computing, a concern is the creation/improvement of European industrial capabilities in the Cloud market at global scale. While cloud provision is dominated by major US firms, Europe is characterised by smaller suppliers which generally provide software services to public and business clients. To strengthen the European Cloud system, Europe should devise and implement a range of policies which need to take this difference into account. Short term research should focus more on facilitating interoperability to address the heterogeneity of offerings, markets and solutions, rather than looking for answers to issues already satisfactorily solved⁴. In addition, the uptake of cloud services should be encouraged to support innovative SMEs and facilitate the creation of large scale European providers.

3.3 High Performance Computing

The next frontier of High Performance Computing will be the design of exascale computing systems by end of this decade, an increase of 1000X in performance over present HPC systems. Although such an increase has been met in the past when moving from terascale to petascale, current software technologies are not anymore suitable for the forthcoming high-end HPC systems due to their high degree of concurrency required to reach the exascale performance. It is estimated that such systems will need between 100 million to 1 billion cores and will consume 100 MW. Although dedicated hardware, such as the new generation of low power consumption processors, might help to keep the energy envelope within reasonable bounds, a tight cooperation between the hardware and the software stack, including the operating and runtime systems as well as the applications, is necessary to take up the energy challenge. Programming exascale machines represents another challenge due to the complexity of these machines: core heterogeneity (CPU & GPU); deep memory hierarchy (cache, local memory, remote memory); and communication and synchronization mechanisms (messages, shared memory). New programming models must be invented providing a high level of abstraction while allowing application programmers to focus on their research fields rather than managing low-level technical aspects, often dependent on the target architecture. Such programming models will have to ensure “performance portability” on the whole range of exascale systems as the HPC landscape will not be homogeneous in terms of design choices. Due to the unprecedented number of components in exascale systems, failures are expected to occur many times during the execution of an application. New innovative approaches must be investigated as current solutions, e.g. checkpointing the applications on a regular basis, are unable to scale. Runtime systems will have to handle failures instead of the application programmers. They will have to detect and predict failures, take the necessary decisions to reallocate resources in an autonomous way while preserving performance. Exascale systems will also provoke a data “tsunami” with applications generating tremendous amounts of information. Rapidly storing this data, protecting it, and analyzing it to understand the results are significant challenges at such scales.

Resulting key software technologies:

- Energy-aware runtime systems to maximize performance while reducing power consumption

- New hierarchical programming models combining data and task parallelism and incorporating message-passing and global address spaces in a coherent framework
- Autonomic runtime systems to cope with failures
- Highly scalable middleware-level I/O libraries for storage systems with sustained and predictable performance.
- Interchangeable SW components smoothly in all systems (cross domains), including safety-critical software intensive systems.

**Recommendation 3:** Launch a European initiative on software approaches for advanced computing systems.

**Recommendation 3.1:** Considering the mainstream positioning of parallel processing, potentially impacting the whole software industry, a coherent software ecosystem will be required to manage and hide the heterogeneity of existing and future parallel architectures from the programmers. By 2020, the old paradigm of software programs developed for a single hardware platform or operating system will be more and more obsolete. Programs will be developed without basic knowledge of their runtime targets. Commodity computing systems will range from smart phones to high-end servers, each of these platforms featuring various granularities of hardware parallelism and hardware heterogeneity. New software technologies, that tailor applications to the underlying hardware at deployment and run time, will be required. Performance will not be the only criteria for optimization; reducing power consumption will have also to be considered to increase the autonomy of portable devices.

**Recommendation 3.2:** Considering the peculiarity of the European Cloud industry, EU research policies should address technical challenges like elasticity, scalability and data management systems, and should aim at easing the uptake of cloud services. This goal entails a range of actions like the development of interoperability to address the heterogeneous environment and facilitate the creation of larger cloud service providers, the development of new cloud based software services to enlarge the addressable market, the definition of standards for increased portability. A key point is the development of solutions providing increased security and privacy in compliance with European regulations.
**Recommendation 3.3:** Reaching the exascale level of performance will require an innovative approach to design effective software systems and tools. A closer cooperation is required between hardware designers, software specialists and application programmers. In order to meet this exascale challenge, Europe must launch initiatives, following a co-design approach, to let HPC vendors to work closely, in a tight cooperation, with academia.
4. Key Software Technologies for Building Data Value

Companies, scientific labs, and many other organizations are facing an explosion in the quantity of data that is being produced and the software techniques for extracting value from this data are not keeping pace. As a result, they are missing fantastic opportunities to build data value based on analysis and mining. For example, this may be the case for a web company that is unable to extract, from the buying patterns of its customers, key behavior trends. Another example may be a physics lab that cannot verify a theoretical hypothesis in huge volumes of experimental data.

One can regularly observe applications facing the difficulties of processing large volumes of the following (non-exhaustive) kinds of data:

- Scientific data coming from experiments or simulations;
- Open data and linked data in particular coming from government initiatives;
- Company data, e.g., in supermarkets, telecom, transportation;
- Web data for all kinds of surveillance from business intelligence to climate crisis management.

Besides the volume, there are a number of reasons for the task to be challenging. First there is an issue of diversity. The information to be analyzed may come in a variety of forms from very structured (tables in relational systems, XLS, HTML, PDF) to semi-structured (XML, web services, RDF) to totally unstructured (sound, image, etc.). Another issue is that data may be distributed between large numbers of devices of which many may be mobile. Heterogeneity is a classical source of difficulty with applications targeting sources with different systems and formats, different models and ontologies. There are quality issues with imprecise data, inconsistencies, and incompleteness. Indeed, besides having to evaluate the quality of data, we often have to evaluate the trust we can have in specific sources. This entails carrying provenance information: e.g., when and where was particular data obtained, how. (E.g., in scientific data, data from experiments are totally useless if the conditions of the experiments are ignored). Finally, in many applications (e.g., in the news) we are particularly concerned with the changes and the trends.

Let us briefly describe operations to be performed on the data. Some are very basic and classical data management tasks such as localizing data, searching and querying, monitoring to detect pattern of changes. Data integration, another classical data processing task, brings in new challenges because of the volume, but also because of the varying quality of data. The crux of the problem is data analysis in particular with the support of visualization. One typically has to analyze data with many dimensions, possibly complex aggregations, and uncertain (e.g., probabilistic) nature. The goal is to extract knowledge of interest buried in the masses of data.

Visual analytics (a combination of information visualization and data mining) aims at the close coupling of visualization and automated analysis, enabling the user to evaluate the results and guide the analysis for an iterative refinement of the analysis. Providing this flexibility to the user imposes specific requirements to the techniques and algorithms, and the key software technologies, which are not resolved in the current state-of-the-art. Specifically, this includes techniques for the analysis
of multidimensional, geo-referenced, unstructured data like text, streaming data and — most importantly — the integration of different formats and data types.

Sometimes the questions that are asked are very precise. Sometimes, we simply want to mine the data to discover “something of interest”. The range of knowledge we can obtain is therefore as wide as the range of opportunities that may be brought by their discoveries.

These problems raise a number of key technical challenges:

- Data intensive parallel computing
- Asynchronous computing
- Semantic-based data integration
- Scalable data management approaches including analysis and visualization
- Surveillance of dynamic and mobile data
- Optimization and balancing/delegating work, e.g., moving algorithms to data, not vice versa.
- Interoperability of data, techniques, and methods across tasks and scenarios.

The valley of death in this area results from the fact that often algorithms developed in the laboratory do not scale to real-world data sets. Researchers are unable to access realistic data sets because of issues such as commercial sensitivity, confidentiality and security. Companies are unwilling to share data because of proprietary reasons; however it is likely that increasing economic value will be generated by federation of data from different proprietary sources.

The development of databases in the past has mostly obeyed a principle: universality. One was typically trying to develop generic solutions vs. solutions to very specific problems. There has been a recent trend in view of the complexity of some of the problems to move to application-specific solutions. Because of the number of problems that we are facing, this approach does not scale and we have to reach some form of compromise between the “one-size fits all” approach and “one-size fits too few” approaches. Semantic Web technologies that include new approaches to the management of data and the integration of structured and unstructured data are an important driver here.

**Recommendation 4:** Create a European Data Observatory that builds upon the open data initiatives for the public sector in Europe and extends it to the private sector. This would provide a universally accessible test bed.
5. Key Software Technologies for Social Computing

We will soon be living in a world where billions of individuals will produce and exchange information with billions of devices. Many of these devices will be able to store and process information and extract knowledge from it. We will thus be living in a world where knowledge will be pervasive.

We have to reconsider the notion of data, information and knowledge in this social context. The bulk of raw data and information comes from users. These are individual’s opinions in tweets, blogs, web pages, emails, etc. Individuals may even participate in large-scale data processing via crowd-sourcing and in application building through the new but as yet little understood mechanism of collective intelligence as demonstrated in Wikipedia. There are also large volumes of information that are drawn from machines observing human behaviors: what people read, where they go, what they buy, etc.

We are beginning to feel that we are drowning under a deluge of information but we have only seen the surface of it. We need the full support of machines to survive because the volume of data and the tasks are beyond human capabilities. We need to develop the tools and interfaces to support the interaction with the masses of data surrounding us. We need tools to find data, analyze them and extract knowledge from them. This is essential to guide our decisions so that we can remain in control of our lives.

Perhaps next to the concerns of being able to deal with the volume of data, privacy is a serious worry for individuals. These new services work by gathering lots of data on individuals and sometimes by federating data from different sources. This presents a tradeoff between obtaining better services and risking the leaking of private data. Work on differential privacy has shown the limitations of totally limiting the distribution of private information. This leads to controlling and regulating what specific software does with our personal data and monitoring that data is indeed used properly.

The goal is to develop novel, life-changing services by extracting value from the data that is available. For this, we need to address the following key technical challenges, some of which we already encountered in the previous section:

- Data intensive distributed computing
- Semantic-based data integration
- Reasoning in a huge-scale distributed environment
- Surveillance of dynamic and mobile data

There is an increasing trend to set these issues in the wider context of social computing which is concerned with the intersection of social behaviour and computational systems and the development of what might be called social computers or social machines. This is a new area of research and development that requires input from many different disciplines such as the social sciences, law, economics, cognitive science, and business studies, as well as mathematics and
computer science. It is also very related to the development of open data initiatives and the innovation and entrepreneurship possibilities that arise as a result.

One problem facing researchers/engineers, both in industry and academia and from computer sciences to social sciences, is having access to suitable data sets to suitably test the scalability of their solutions. This is consistent with Recommendation 4 which concerns the creation of a European Data Observatory that will provide access to data that industry (particularly SMEs) or research labs cannot obtain individually. This may be distributed over a number of centers and would include results of web crawls, real time web surveillance and incorporate large corpuses of data from business partners. The goal would be to offer for social or web data, the same kind of environment that is available in scientific computing centers for scientific applications.

**Recommendation 5:** Develop interdisciplinary funding programmes to understand the concepts of social computing/computations, its societal value and the innovation and entrepreneurship possibilities that arise as a result. It is very important to consider privacy, security and trust issues alongside such developments. There is an increasing need for the development of data observatories to support research and development in this area.
6. Key Software Technologies for Internet based applications and real time services

The Internet and, more generally, telecommunication networks and services can be considered as ultra large scale software systems, characterized by their intrinsic distributed nature, large geographic coverage, topological complexity, cross-organization management, high dynamicity. The behavior of such systems is nonlinear and can even exhibit chaotic properties: the software part of each individual component can be mastered by a single actor but the overall architecture can lead to new emerging phenomena that are difficult to understand, model and control. In such a context, maintaining global or cross cutting properties such as security, quality of service, resilience or resource optimization can be a time consuming and high risk process, regarding the dependency of modern societies on their telecommunication and IT infrastructures.

In producing web services, the plethora of programming languages has meant that there is a trend towards mixed language working with tactical, bottom-up choices being made about which language best suits the task at hand. Where one would have recommended a full C++ or a full Java based vertical solution, we could have today a mix of node.js / PHP / iOS / Android / C# to leverage and adapt to a set of existing ecosystems. The open protocols (SQL, JSON, HTTP, SSL, etc) are solid and rich enough to allow this technological fragmentation. The consequence is that the expectation for code reusability decreases, while the adaptability skill of the developers is more and more crucial. For instance, whilst there is no need to focus on power consumption for every new service, it can become critical for the few services that turn to be massive success: this necessitates the ability to change used technologies while the service is live. This requires a radical re-think about how providers approach design.

Tools are also becoming more important: there are huge assets available, usually for free and more and more under friendly terms and conditions. SDKs (editor/compiler/debugger) are now just a small part of what is needed: collaborative working tools; a modern source code repository; build and integration frameworks; testing robots; static code analysers; bug management solutions; release tools are all necessary and available for free (or almost for free). Thus access to the technology is not a competitive advantage anymore. However, the ability to plug the latest and most efficient tools in a consistent and end to end discuss/design/develop/test/deploy environment make a clear difference. In particular the ‘DevOps’ paradigm allows faster develop/release cycles than ever. For Telcos, it’s again a change because they used to design internally, outsource development, and manage again releases by themselves, making the ‘DevOps’ approach difficult to integrate. The successful operators in 2020 will probably be the ones who adapt best to this trend.

The third and last trend is an opportunity for Telcos: the recent trend in devices and web services, with a clear independence between software and hardware, should happen for network infrastructures as well. Virtualization (of course), management of multi-core in the OS kernels and development tools (for performance), web standards maturity (for management) will be central technologies to lower the capital expenditure on one hand and allow quicker cycles on the other hand. Open source as a general way to distinguish commodities from added value components should increase these benefits, by encouraging convergence and maintenance cost sharing on non-differentiating elements.
New software technologies induce a new way of working. HTML5 is a good instance: the technology by itself is not so disruptive, but the players behind it are not the same and their way of thinking about software is completely different. Telcos will need to handle both aspects.

Some key technical challenges are:

- modeling the architecture and behaviours of large scale distributed software systems
- predicting their dynamic evolution and the potential emerging phenomena and develop the associated engineering tools
- designing a way to monitor and operate such system spanning different actors and organization
- automating operations to implement real autonomous system able to cope with changes (growth, failures, new usages, ...)

Telecommunications are still implementing a fundamental paradigm shift from hardware equipment implementing well known and standardized protocols to general purpose computers executing distributed applications relying on constantly evolving interfaces and data formats. Even high performance dedicated devices such as routers or access multiplexers move to architectures based on a low level efficient resource management layer (fast path) controlled by a programmable and open software layer (slow path). Moreover edge customer premises equipments (computer, connected devices, smart phones, ...) can be considered as additional software platform with equivalent storage of computing capabilities. Keeping consistency across these layers up to the applicative and even semantic layers requires new tools, both in the design phase and in the deployment one.

In this area, some key technical challenges are:

- software optimization and certification based on semantic modeling and formalization of requirements and implementations
- separation of concerns and late reconciliation of functional and non-functional properties, in multi-layered architectures
- information centric networking, content aware networks
- coexistence of multiple software levels, each using their own frameworks, languages, processes.

The sheer complexity of modern Telco systems and the myriad of services they host create a cognitive gap between the architectures and behaviors of such systems and the understanding and capabilities of the human being either creating, administering or simply using these systems. In addition to R&D on the key technologies, Europe should lead the studies on cognitive psychology and social science necessary to understand individual and social behaviors in such hybrid human-software systems. Key research could be performed, inter alia, on cooperative software design or system operations; task oriented software design and human-machine interfaces; information overload and knowledge based systems; and ontology based systems.

We need to reach a state where software for real time service systems, e.g. telecommunication systems, is predictable in different configurations, not least to deal with updates decades after the product was delivered. Ideally, later date product tests should not be needed since the quality of the
integrated variants can be deduced from verification on component or subsystem level. It shall be able to predict the behaviour of systems with many different components, especially in real-time systems. Security and integrity aspects should and could become part of any system development.

Another growing market is defined by the Internet of Services as addressed in the Future Internet PPP and other major European lighthouse initiatives. These business services will build upon the above mentioned web services but require further functionality such as contract negotiations & management, service orchestration, and composition based on business model harmonization. Here, a worldwide standard suite is missing to complement the above mentioned Internet standards. Europe should not miss to define such a set of standards.

**Recommendation 6.1:** Support the effort that by 2020, software intensive real time systems should be executable on shared hardware and easily connectable to the outside world. Issues of flexibility, security, integrity, portability and migration must be natural ingredients of any software development ecosystem.

**Recommendation 6.2:** Europe should coordinate the standardization of the IoS and with the support of the FI PPP create/drive the worldwide standard suite for the service “prosumption”.
7. Key Software Technologies for Embedded Systems

Embedded Systems provide intelligence to physical objects of everyday life (i.e. to products/artifacts/systems like cars, aircraft, trains, personal devices, medical devices, industrial plants, power plants, etc.), making them smart objects. In the next years, by further exploiting the Internet and the world-wide-web, embedded systems will increase the intelligence, control and communication capabilities of a wide range of objects, enabling their interaction and cooperation with people and organizations. Such smart objects will be joined together to create highly distributed systems, called Cyber Physical Systems, able to dynamically grow and adapt to the needs of individual users and communities. This convergence will bring a wealth of opportunities and innovations in technology, applications and business models. For instance, innovative products, services and solutions will be created to respond to key societal needs through accurate and continuous real world knowledge. As a consequence, the potential to innovate of industries throughout the world will be influenced to a large extent by their capability to leverage the sophisticated technical solutions provided by the future embedded systems.

Embedded computing has been traditionally an area of strength for the European ICT sector. Europe’s strength is based on two different factors.

The first is related to the wide range of application domains for which ICT is emerging as a key driver of innovation and where Europe is a world industrial leader. Such domains include transportation (aeronautics, automotive, rail and maritime), healthcare, energy (production and distribution), telecoms and security.

Strong tool offering by a number of software editors and strong in-house system design and development capabilities characterize European integrators of embedded systems, facilitating also the transformation of research into products and services in line with market needs.

The second factor concerns a number of technical areas where Europe has enjoyed world leadership, like embedded and distributed software, hard real-time design, dependable computing systems, software agent technologies, system of systems design.

As a consequence of the breadth and depth of the embedded systems sector in Europe, the economic impact of embedded systems on European industrial success is therefore huge and it will increase in the future, especially in the area of Cyber Physical Systems.

Technological advances in embedded systems will help transform our world with systems that

- respond faster to the environment (e.g., autonomous collision avoidance),
- are more precise (e.g., robotic surgery and nano-tolerance manufacturing),
- work in dangerous or inaccessible environments (e.g., autonomous systems for search and rescue, firefighting, and exploration),
- provide large-scale, distributed coordination (e.g., automated traffic control, smart grid),

---

5 Recommendations of ISTAG on FP7 ICT Work Program 2013, March 2012
6 ISTAG, Shaping Europe’s future through ICT, March 2006
8 ISTAG, Shaping Europe’s future through ICT, March 2006
are highly efficient (e.g., zero-net energy buildings),
- augment human capabilities, and
- enhance societal well-being (e.g., assistive technologies in the house and ubiquitous healthcare monitoring and delivery).

These capabilities will be realized by deeply embedding computational intelligence, communication, control, and new mechanisms for sensing, actuation, and adaptation into physical systems with active and reconfigurable components and their connection and interoperation by global digital networks. This progress will transform how we interact with the physical world just like the Internet transformed how we interact with each other.\(^9\)

Far from being static, ICT is developing more quickly than ever because of technological progress and the massive technical and financial resources deployed worldwide. The new trajectories of ICT however might not favor Europe. As noted by ISTAG\(^10\): “In the last few years, the introduction of large server farms with more than 1,000,000 computers comprising huge data storage facilities, connected to the Internet, and the development of software for the nearly autonomous management of this multitude of computers has created a computing infrastructure that has improved the price/performance ratio by one or two orders of magnitude over the price/performance of conventional data centers. Such an economic advance will have a profound impact on the future computational infrastructure. This future infrastructure is expected to consist mainly of smart personal devices and embedded systems at one end of the Internet and a diversity of server farms (the cloud) at the other end”.

While the convergence between embedded systems and the Internet will open up huge opportunities for the ICT system sector in Europe, neither smart personal devices nor the cloud are among Europe’s strengths. Furthermore, the main actors in the Web are not European and they have already fully grasped that the future generation of ICT will be networked, mobile, embedded into everyday life, based on a rich and multimodal interaction (see for instance Google’s recent Glass project). Therefore, the competitive environment looks increasingly hostile. Europe will need to develop new technologies, applications and business approaches if it wants to remain at the forefront of ICT innovation and to address, through ICT, its societal challenges.

For the embedded systems and cyber physical systems sector, technological progress will mainly concern: managing the integration of computing with increasingly complex physical processes; addressing real time system complexity; developing efficient and accurate simulators of the physical systems to be observed and controlled; exploiting the ubiquity of broadband mobile connectivity; managing the growth of intelligence “at the edge”; maintaining safety, dependability and security of use; minimizing energy consumption of embedded devices; exploiting storage and processing power on-demand; enriching content, user experience and performance through the convergence/integration of applications, processes and services; and creating new user-friendly interfaces able to talk to a very wide variety of objects and to manage 3D, multimodal and geo-referenced data.

---

\(^10\) Recommendations of ISTAG on FP7 ICT Work Program 2013, March 2012
There is also a need in Europe to overcome fragmentation and to progressively build on research results to feed excellence in innovation. The aim is to reduce development and PLM (Project Lifecycle Management) costs, shorten time to market, correct poor productivity and increase affordability while continuously improving performance. However, all the above will not be enough for Europe to bring to market all the new opportunities which will emerge in this decade in terms of breakthrough services and applications. Successful exploitation of ICT research results will require also increasing the research effort in academia and industry, nurturing entrepreneurship and the creation of new businesses and business models, and improving the permeability between academic research and industry. In other terms, we need to overcome Europe’s innovation gap.

**Recommendation 7:** Embedded systems will increase the intelligence, control and communication capabilities of a wide range of objects, enabling their interaction and cooperation with people and organizations. Such smart objects will be joined together to create highly distributed systems to address for instance societal challenges. To support this evolution, R&D for Embedded Systems should be extended towards the system of systems area leading to Cyber Physical Systems. As a consequence, Europe should develop new scientific foundations, system design methodologies, development processes and tools to create the technical solutions tackling the challenges posed by system complexity (e.g. system behavior, dynamic growth, availability, fault tolerance, safety and security).
8. Key Software Technologies for Human-Centred Computing and Multimedia

8.1 Human-Centred Computing

Although information technology touches every aspect of our lives, much software technology remains mired in a reductive “human-in-the-loop” paradigm. We need to shift our perspective, emphasizing a “computer-in-the-loop” approach that enhances human activity, placing users squarely at the centre of the design process. The goal is to create software technologies that empower people while avoiding unnecessary constraints and complexity. Human-Centred Computing combines the use of computers as tools that augment human capabilities (Human-Computer Interaction), as partners that provide intelligent support (Artificial Intelligence) and as a medium for human-to-human communication (Social Media).

Human-Centred Computing must integrate user-centred design methods into software design workflows, with usability as an essential metric for assessing quality. Interactive software cannot be measured solely in terms of machine performance, but must assess human performance and enjoyment as well. Beyond usability, examples like the iPhone or iPad show the business importance of an attractive user experience design that increases emotional product value for customers. A second goal is to increase accessibility, not only for people with disabilities but also for the general public. Software technology must support users who view technology not as an end in itself, but as a means to accomplish their goals. Third, we need software infrastructures and guidelines for usable privacy, so that users can understand and control the information collected and communicated about them, including a “right to forget”. Finally, adaptability, context awareness, and individualization across ensembles of devices are of high importance. We can expected that, on the one hand, communities of users interact over the same data from different perspectives and using different types of devices while, on the other hand, a single user may interact with an IT-infrastructure across different locations, devices, and contexts.

These overarching goals affect many key technologies and application areas described elsewhere in this document. However, significant advances will require focused research and development in fundamental aspects of Human-Centred Computing, including human-computer interaction, interactive visualization, multimedia computing, multimodal interfaces and ubiquitous computing. Key software technologies include:

- **Novel interaction technologies** that enlarge the bandwidth of human-computer interfaces, offering users greater forms of expression through, e.g., advanced gesture, voice and brain-computer interaction. These technologies must better sense and interpret the state and dynamic nature of users and their environment, and must generate more understandable, interactive representations of the changing state of the computer system.

- **Software toolkits and languages** that embody human interactions as first-class objects and support mixed-initiative approaches in which users and computers collaborate towards a shared goal. Although current approaches were appropriate for automating operational
tasks, supporting creative and open-ended activities requires interaction as a central concept.

- **Interoperable interaction technologies** that provide users a seamless interactive experience involving multiple devices and services. This requires a fundamental reassessment of today’s interaction paradigms, which were based on single-user, single machine assumptions.

- **Human-centred design thinking approaches** that provide an attractive user experience to improve emotional customer value of software-intensive systems.

**Recommendation 8.1:**

- Human-Centred Computing must integrate user-centred design methods into software design workflows with **usability** and attractive **user experience**, increase **accessibility** and integrate **usable privacy primitives**.

- Research has to be undertaken towards the integration of cloud services, device ensembles, and community-based interaction, leading to the paradigm of "user-centred interaction in the cloud"

- New programming paradigms and patterns have to be developed that support different areas of human-centred computing such as self-adaptation, individual configuration of ensembles, artificial intelligence assistance, multi-media content and multi-modal interaction

**8.2 Multimedia**

Actual Multimedia Technologies are driven by three trends. User-driven Internet portals like Flickr or Youtube reach extensive usage (for YouTube alone, 2 billion views a day and additional 24hrs of video footage being uploaded every minute). In addition, mobile systems like smartphones and tablets combine a powerful computing performance with multimodal sensory data. Also, the data is better structured and typically geo-referenced with the latter not only based on 2D maps but also on 3D representations of our real world (e.g. Google Streetview) as reference models.

Within the next decade these trends will merge and in this context 3D and multidimensional multimedia data will play a major role. Thus, multimedia data will not only be linked to real world objects/situations and persons, also multimedia data will be used as virtual representation of our real world. In this sense, real world objects and persons will be linked to their virtual counterparts documenting related information, development in time, functionality and user experience. All this will be realized in user-driven platforms, thus the borders between producers and consumers will blur.

From the technological point of view these trends are driven by broadband network connections for mobile systems. The mobile device (lifetime of backend devices has to increase because of
limitations in material resources) will only act as capturing/displaying console that allows to access computing resources offered in the network. In this regard displaying, sensory, and capturing functionalities on mobile consoles will increase (starting with projection functionalities and range cameras integrated in smartphones and reaching to retina-integrated displays and brain interfaces) while processing is supported only in network resources.

Many users of smartphones and other mobile devices will accept a permanent transmission of their sensor data to the network where sensor fusion will generate a quite detailed view of our real world and in particular of our human society in real-time. This data will be used for accurate real-time simulations supporting intelligent traffic control, accurate weather and air pollution simulations as well as a permanent update of 3D models of our virtual earth which, consequently, can then even describe environments with daily changing geometries like e.g. construction sites.

In this trend, mobile sensory establishes the correlated digital reality as a key technology which requires real-time capturing, processing, simulation, and registration technologies for multi-dimensional data (3D/4D/nD). Multi-dimensional content from individual capturing technologies through active end nodes (this might range from smartphones to cognitive robotics actors) will be widely available and made accessible and distributed through platforms at future Internet level.

**Recommendation 8.2**: Simulated Mixed Reality as enabling technology for a tighter coupling of reality and virtuality seeks for new solutions within the area of real-time numerical and optimization solution schemes, functional simulations, seamless rendering, analysis-through-synthesis and synthesis-through-analysis techniques. The overall vision should be to create perfect (virtual) assets being seamlessly integrated into reality.
9. Key Software Technologies for Enterprise Application

Enterprise software is a key driver of innovation in today's business environment. The dynamic integration of business partners is of ever-growing importance in order to quickly react to customers’ needs. A wide range of different services is increasingly offered and used in the Internet via cloud platforms in a cost-effective and efficient way. The new paradigm of “emergent software” is characterized by combining a wide range of components from different service providers in a dynamic and flexible manner, in order to fulfill the highly complex requirements of digital companies, i.e. companies that have digitalized all of their business processes so that enterprise software becomes the decisive driver behind product and process innovation.

Enterprise software needs to adapt itself dynamically to the changing requirements of the market as well as the business environment and supports service relationships between companies. The intelligent linking of existing offers and the addition of supplementary components and services allows a completely new range of offers to be provided. Companies that make use of such software will be able to realize new business models themselves – models that would have been unthinkable during the original development of the components and services. Users of emergent enterprise software receive individual solutions "from a single source" – via the simple composition of standard solutions from multiple providers. This means emergent enterprise software will be relatively easy to maintain and, furthermore, will not require significant investments on the part of the user. Users will therefore be able to increase the flexibility of their (new and innovative) business models, operate them in a more market-oriented manner, and map them within the software systems.

In order to provide these capabilities of enterprise software, the current state-of-the-art with respect to Internet of Services and Cloud Computing must be extended allowing for semantically connecting software services across enterprise domains. As of today, there are no suitable software engineering methodologies in place nor exists any appropriate standard.

The last few years have already shown an increase in business processes with emergent properties. However, they can so far only be realized with great effort and a lot of manual labour. Especially those companies that are providing Internet-based services have dealt with this phenomenon, e.g. the integration of social networks for quality control purposes of online shops in the fields of travel, books or electronics. Even a traditional sector like the automotive industry has recognized that emergent business processes are required for the positioning of their products.

Further research needs in order to fully realize the paradigm of emergent software lie in the following fields:

Automated Interoperability: In contrast to self-organizing systems there is no need in the B2B domain for ad-hoc networking in the sense of a spontaneous aggregation of services. Rather, the existing services need to be linked efficiently and quickly to new business processes without new interfaces having to be implemented. New services have to be designed in such a way that they are applicable and adaptable in unfamiliar contexts. The key here lies in the automated production of semantic interoperability and in the generation of required adapters, but also in mechanisms for searching, finding, evaluating and configuring components.
Automated adaptivity: new and innovative business models require a new class of adaptive services and components. They must adapt dynamically to the current circumstances. The difficulty, however, lies in the cross-boundary context of two or more companies involved where no end-to-end view of the business process is possible. In addition, a cross-organizational analysis of the current system behavior and the resulting recognition of a need for adaptation is required.

The resulting adaptation has to be automated and performed in compliance with SLAs. Through the use of components not known beforehand, these properties have to be taken into account at the time of development.

Recognition of user context and guidance: Innovative business models require a context-sensitive presentation of information to the user. This requires knowledge about the current situation of the user, the context and the resulting Information needs. A further requirement is to provide a homogeneous look-and-feel as a variety of different devices or device classes is supported and the services come from different providers. Future business models will be based on online-communication of communities of experts with diverse experience as well as the support of customers with a wide range of context-specific requirements. This implies the need for appropriate user interfaces and visual computing techniques (see Chapter 8).

Tailored Security: Cross-company emergent business models can only be successful if they are trusted. This implies that they have an adequate level of security. Important security features include the ability to identify components and their properties to reliably verify the correct functioning of services, while protecting the privacy of individuals, organizations and businesses properly. The properties of components and services and the protection goals for the data must be specified in order to enforce the appropriate security features at the various levels of the system. Particular attention should be paid to usability. The technical implementation of security requirements must be complemented by social processes that support trust.

Massive real time main memory data analytics: The Internet of Things will lead to connect more than a trillion devices according to IBM Research to the Internet. The data points generated will lead to explode current database technology and business intelligence. The rising availability of main memory database systems and the use of them for real time analytics lead to a complete new field of real time analytics not possible with today’s enterprise software architecture. While first technologies are available, the related programming and design support is very limited and does not allow for easy adoption. New software development methodologies as well as tools could open up a domain considered so far as “none real time” analytics. Particular focus should be given multimedia analytics, in particular 3D data.

Many Core Programming: Enterprise software has been largely developed using the three tier client server paradigm as underlying architecture. Thus, SOA-based systems still are largely performed in a sequential way not leveraging the increasing capabilities of many core systems. Recent efforts on using main memory database concepts have demonstrated significant performance improvements over and above the classical database server approach. Combining this approach with parallel programming could generate a complete new generation of enterprise software systems coping with the increasing amount of raw data in a completely disruptive way. As of today, there are neither suitable programming languages nor development technologies to serve the development of
massive parallel, main memory based enterprise software systems. Current research is promising but requires a significant push beyond the current rather tactical approach.

**Recommendation 9:** Develop a European strategic initiative on enterprise software technology to maintain Europe’s leadership. This initiative should stimulate to develop the next generation disruptive enterprise software technology and related worldwide standards. The world is facing the start of a new era of enterprise software given the ground breaking technologies such as cloud computing, many core, main memory database technology are all now available. In addition, the Internet of things has become reality generating a volume of data never envisioned in mankind history. Enterprise software needs to cope with this massive data to provide business process management taking real time real world information into account.
10. **Key Software Technologies for Ensembles – the future generation of software intensive systems**

In the previous chapters (3 through 9), we have discussed key software technologies for a variety of application domains within computer science. Starting from parallel computing and cloud computing, we have discussed data value, software for the social world, web applications, embedded systems and sensor networks, internet and telecommunications, and finally multimedia and enterprise applications. The key software technologies and research challenges identified in these chapters stem from the specific characteristics of the software systems required for each domain in 2020. However, taking a broader perspective, we can identify several general qualities which are common to all or most of the systems identified in these chapters, and thus arrive at several overall technical challenges to be investigated.

A first key characteristic of the future generation of software-intensive systems in 2020 is the **massive number of nodes per system**, which will be one of the most visible features. This is true not only for distributed systems consisting of nodes ranging from mobile phones to high-end servers, but also for individual nodes with the advent of multi-core processors, requiring support for parallelism and scalability (chapters 4, 9). Furthermore, the availability of cheap, low-energy mobile devices ensures that we will see an increasing number of elements with computational capability in the next years. Smart grids, automatic traffic control, and the sensor networks built by mobile devices such as phones are examples where large-scale distributed coordination is required (chapters 6, 7). Distribution not only affects functionality, but data (chapters 4, 5) – data must be integrated and there is a need for reasoning in huge-scale distributed environments.

A second area of interest is **adaptability**. Future systems will often have to operate under conditions that differ significantly from the ones for which they were designed. They should not only be able to cope with changes in their environment (growth, failures, new usages, …) (chapter 6), they should also be able to work reliably in the face of modifications to their execution platform, for example with the help of autonomic runtime systems (chapter 3) which provide automated adaptability (chapter 9). Adaptability is also required in autonomous systems for search and rescue, firefighting, and exploration – in general, dangerous or inaccessible environments (chapter 7). A related challenge is predicting the dynamic evolution and potential emerging phenomena (chapter 6).

In order to provide adaptability, system components must be able to understand one another, which is being complicated by the expected continued **heterogeneous** nature of future systems. We are moving to a situation where software is infrastructure, in which we have a coexistence of several layers, each using their own frameworks, languages, and processes (chapter 8). This technological fragmentation (chapter 6) is only possible when we employ open standards. Interoperability is even more important on the data level, where a key enabling technology is semantic-based data integration (chapters 4, 5).

We call the future generation of software-intensive systems with the above characteristics **ensembles**. Correspondingly, **ensemble engineering** is the science and engineering discipline of complex, integrated ensembles of computing elements. The potentially huge impact—both positive and negative—of ensembles means that we need to understand ways to reliably and predictably
model, design, and program them. Because of the social and economic importance of ensembles, the difficulty of building them, and the lack of engineering methods that adequately address the challenges posed by future software-intensive systems, we consider ensemble engineering to be one of the most important areas for European ICT research in the coming years.

We can identify two fundamentally different kinds of ensembles with regard to their connection to the outside world.

Firstly, societal ensembles are ensembles that are closely connected to humans. At the forefront of such ensembles are mobile devices which directly collect data from the user and have access to her reactions; however, user-driven internet portals are another source of data (chapters 5, 8). This raises privacy questions on a European scale (chapter 3), in particular with regard to the surveillance of dynamic and mobile data (chapters 4, 5). Other aspects of societal ensembles are recognition of user context and guidance (chapter 9) and providing a correlated digital reality (chapter 9) or otherwise enhancing human capabilities (chapter 7). Societal enterprises can also enhance well-being for example by ambient assistance or ubiquitous healthcare monitoring and delivery. Research in this area will have to investigate the dynamics of purposive interactions and how the structure of evolving societal interactions can be reflected in the architecture and the design of the software: evolution of societal ensembles has to be a long-term process that goes beyond single-run adaptation, and systems have to maintain societal coherence while supporting diversity and context awareness.

Secondly, we have physical ensembles, which are intimately connected to the physical world in space and time. They are equipped with sensors and actuators and have to take into account issues of locality and resource constraints. As mentioned above, the individual nodes in such a system may be smartphones with sensors for traffic control, air pollution, or similar (chapter 9); they may also be directly embedded into vehicles, robots, or houses. Key areas of interest here are a fast response to the environment (for example for autonomous collision avoidance), precision work (for example in robotic surgery), and energy efficiency (chapter 7). Coordination in space and time with limited resources is one of the major challenges faced by physical ensembles.

Many of the problems of ensemble engineering are directly connected to the massive scale of ensembles, the complexity of its components or their interaction, the operation in open environments, and the need for adaptation: How do we design software for ensembles that is reliable, predictable, with guarantees for security and trust, that acts autonomously, has self-* properties (self-healing, self-managing, etc.), and that can harness emergent behavior? The research effort must span the range from foundational topics to the construction of test beds and systems to verify the proposed methods:

- **Foundations, modeling, analysis.** Massive scale, complexity of components and interactions, heterogeneity, and adaptation are properties that arise directly from the definition of ensembles. Yet our current understanding of system and software engineering is not sufficient to reliably build software-intensive systems of the required size and with the desired properties. We need formal foundations as well as modeling and analysis approaches that allow designers to build reliable, trustworthy ensembles

- **Languages, compilers, and platforms for ensembles.** It is not sufficient to build the foundations for modeling and analyzing ensembles, it is also necessary to provide engineers
with the languages, compilers and platforms to build them. Languages and platforms need to support reflection and dynamic evolution of ensembles; models of the ensemble have to be available at run-time. Research into languages, compilers and platforms for ensembles is an important foundation for building adequate tools for ensemble engineering.

- **Methods, tools and processes for ensembles.** Our current methods and tools are barely adequate for dealing with current systems. Even the best available tools and languages are only slight improvements over the state-of-the-art from 30 years ago. Development processes and methodologies are mostly based on conjectures and opinions, without scientific studies backing up the claims made by their proponents. Current development tools provide only limited support for distributed development and make no use of the advances in cognitive systems and ambient computing environments. Processes for engineering security-critical ensembles will probably need to include formal methods for reasoning about ensembles that are scalable to the size of real-world ensembles, as well as methods for controlling and adapting ensembles whose subsystems and components are dynamically changing. This may necessitate finding replacements for the notion of “correctness according to a specification” which are more appropriate for the scale and complexity of ensembles. Research about development methods, tools and processes that provide drastically improved support for the developer is sorely needed.

Test beds and research prototypes for both physical and societal ensembles should be built as part of future research efforts. Research on ensembles can be performed in different directions: one method is to use a bottom-up approach that starts with logical foundations, another method is to use a case study-driven research plan that starts with the construction of a prototype and develops the necessary tools and formalisms based on the experience gained from the development process. The most promising research agenda may be a combination of these approaches, where both bottom-up and top-down approaches are combined, and the development of foundations, languages, tools and processes is informed by, but not completely determined by, case studies.

**Recommendation 10:** Set up a FET Flagship to support the right timescales, levels of ambition and long-term funding that would allow Europe to maintain its pre-eminent position in the future generation of software intensive systems.
11. Conclusion and recommendations

Software is a key driver for the European economy. Whilst the European software industry lags behind its international competitors, recent trends in technology (for example Future Internet, Mobile Computing and Cloud Computing) offer a window of opportunity to rectify this situation. The commoditisation of software and infrastructure inherent in the move towards Cloud Computing and the continued advances in Open Source and mobile applications should help to foster greater innovation. Europe must ensure that the younger generations are equipped with the appropriate technical and business skills to exploit these opportunities. Non-functional features such as reliability and security will become more important differentiators. Increasingly software will be automatically produced and Europe needs to ensure that the appropriate production methods are employed to ensure that the European software sector is market leading in these aspects.

The old approaches to software development are simply not sufficient for maintaining global leadership. To remain a strong competitor, the European software intensive industry must increase its efficiency and productivity, enhance the software quality and reduce lead times. Software technologies are also key in social interactions, not only for simulating these interacting but also for enabling new interacting modes. Web technologies and social networking are one of the most spectacular technological advances in the society evolution. This soft (r)evolution ends the information era and opens a new era : the digital society. We propose the creation of a European Software Agenda which could incorporate a long term research agenda coupled with an innovation agenda and the creation of new business models.

11.1 Main recommendation

A Strategic Agenda for Software Technologies in Europe should be created in cooperation with Industry, Academia and Public sector. The agenda should outline the strength from a European perspective and how we can renew and strengthen it. Common goals and needs in the short, medium and long term should be described. It should also describe what actions should be implemented to achieve these goals and needs and how we can, in a smarter way, make use of existing efforts, resources and facilities. The agenda should also draw up proposals on how the strategic work should be organized, run and quality assured. The first version of this Agenda should be delivered before mid-2013.

11.2 Software Engineering

Software engineering goes far beyond coding and includes topics such as theoretical foundations for languages, models and systems, requirement handling and understanding, architecture design, system design, integration, specifications, documentation, verification, test, software quality,
security-by-design and other higher-level activities. Creating brand new programs is a rare task and software engineering has to focus less on creating systems from scratch and instead emphasize techniques and tools for extending and changing existing systems. Universities need to focus more on software engineering as an academic discipline to develop and accredit graduates who possess the wide set of skills necessary to successfully build software: strong theoretical foundations paired with a willingness to grasp industrial demands on complexity, efficiency and quality, as well as social, linguistic and management abilities. Industry, academia and relevant government bodies should handle this particular challenge with a new attitude.

Recommendation 11.2.1: The scale and duration of the development of existing and new methods, tools and theory is not well supported by the recent research culture, and even less by the funding culture. It will require collaboration of the researchers with the directors of funding policy to outline a strategy for culture change, leading particularly towards larger-scale and longer-term collaborations. The more mature branches of science have shown the way. The role of scientific competition against objective and realistic criteria must also be exploited.

Recommendation 11.2.2: A better environment for understanding the consequences of different implementation alternatives (e.g. quality, robustness, performance, maintenance, evolvability, ...) would also be of great value in answering questions about what should be developed in software, in hardware or as a service. We need to ensure working interchange of SW tools as well as methods and tools for dealing with risks and effective recovery from failure situations in software projects. Europe should set up a directory on good practices and exemplar educational programs for example on energy efficient software, code optimization, best in class software tools, large scale efficiency management,… Europe should foster methods and tools for improved software engineering project management, risk assessment, quality control, and problem recovery in a coherent manner between the technical, the user, and the management perspectives.

11.3 Parallel, Cloud and High Performance Computing

Parallelism has become a mainstream technology in the development of nowadays computing systems impacting the whole spectrum of the computing landscape: from personal systems (smartphones, tablets, PC) to large-scale data centers and high-performance petascale computing systems. It is the only alternative to significantly augment the performance of computing systems while providing better autonomy for small devices or keeping the energy footprint at a lower level for data centers and HPC systems. This evolution has fundamental impacts on software technologies because parallelism has to be managed by software rather than hardware. Europe needs to develop the key software technologies to exploit the quintessence of modern processors, Cloud systems and high-end HPC systems.
**Recommendation 11.3.1:** Launch a European initiative on software approaches for advanced computing systems.

**Recommendation 11.3.2:** Considering the mainstream positioning of parallel processing, potentially impacting the whole software industry, a coherent software ecosystem will be required to manage and hide the heterogeneity of existing and future parallel architectures to the programmers. By 2020, the old paradigm of software programs developed for a single hardware platform or operating system will be more and more obsolete. Programs will be developed without basic knowledge of their runtime targets. Commodity computing systems will range from smart phones to high-end servers, each of these platforms featuring various granularity of hardware parallelism and hardware heterogeneity. New software technologies, that tailor applications to the underlying hardware at deployment and run time, will be required. Performance will not be the only criteria for optimization; reducing power consumption will have also to be considered to increase the autonomy of portable devices.

**Recommendation 11.3.3:** Considering the peculiarity of the European Cloud industry, EU research policies should address technical challenges like elasticity, scalability and data management systems, and should aim at easing the uptake of cloud services. This goal entails a range of actions like the development of interoperability to address the heterogeneous environment and facilitate the creation of larger cloud service providers, the development of new cloud based software services to enlarge the addressable market, the definition of standards for increased portability. A key point is the development of solutions providing increased security and privacy in compliance with European regulations.

**Recommendation 11.3.4:** Reaching the exascale level of performance will require innovative approach to design effective software systems and tools. A closer cooperation is required between hardware designers, software specialists and application programmers. In order to meet this exascale challenge, Europe must launch initiatives, following a co-design approach, to let HPC vendors to work closely, in a tight cooperation, with academia.

### 11.4 Building Data Value

All organizations are facing an explosion in the quantity of data that is being produced and the software techniques for extracting value from this data are not available. As a result, they are
missing fantastic opportunities to build data value based on analysis and mining. Data may be distributed between large numbers of devices of which many may be mobile. Heterogeneity is a classical source of difficulty with applications targeting sources with different systems and formats, different models and ontologies. There are quality issues with imprecise data, inconsistencies, and incompleteness. Besides having to evaluate the quality of data, organizations often have to evaluate the trust they can have in specific sources.

The valley of death results from the fact that often algorithms developed in the laboratory do not scale to real-world data sets. Researchers are unable to access realistic data sets because of issues such as commercial sensitivity, confidentiality and security. Companies are unwilling to share data because of proprietary reasons; however it is likely that increasing economic value will be generated by federation of data from different proprietary sources.

**Recommendation 11.4:** Create a European Data Observatory that builds upon the open data initiatives for the public sector in Europe and extends it to the private sector. This would provide a universally accessible test bed.

### 11.5 Social Computing

Social computing is concerned with the intersection of social behaviour and computational systems and the development of what might be called social computers or social machines. This is a new area of research and development that requires input from many different disciplines such as the social sciences, law, economics, cognitive science, and business studies, as well as mathematics and computer science. Billions of individuals will produce and exchange information with billions of devices, able to store and process information and extract knowledge. It is also very related to the development of open data initiatives and the innovation and entrepreneurship possibilities that arise as a result.

Privacy is a serious worry for individuals. New services work by gathering lots of data on individuals and sometimes by federating data from different sources. Hence a tradeoff has to be found between obtaining better services and risking the leaking of private data. This leads to controlling and regulating what specific software does with personal data and monitoring that data is indeed used properly.

**Recommendation 11.5:** Develop interdisciplinary funding programmes to understand the concepts of social computing/computations, its societal value and the innovation and entrepreneurship possibilities that arise as a result. It is very important to consider privacy, security and trust issues alongside such developments. There is an increasing need for the development of data observatories to support research and development in this area.
11.6 Internet based applications and real time services

Telecommunications are implementing a fundamental paradigm shift from hardware equipment implementing well known and standardized protocols to general purpose computers executing distributed applications relying on constantly evolving interfaces and data formats. Even high performance dedicated devices such as routers or access multiplexers move to architectures based on a low level efficient resource management layer (fast path) controlled by a programmable and open software layer (slow path). Moreover edge customer premises equipments (computer, connected devices, smartphones, ...) can be considered as additional software platform with equivalent storage of computing capabilities. Keeping consistency across these layers up to the applicative and even semantic layers requires new tools, both in the design phase and in the deployment one.

**Recommendation 11.6.1:** Support the effort that by 2020, software intensive real time systems should be executable on shared hardware and easily connectable to the outside world. Issues of flexibility, security, integrity, portability and migration must be natural ingredients of any software development ecosystem.

**Recommendation 11.6.2:** Europe should coordinate the standardization of the IoS and with the support of the FI PPP create/drive the worldwide standard suite for the service “prosumption”.

11.7 Embedded Systems

Embedded Systems provide intelligence to physical objects of everyday life making them smart objects. By further exploiting the Internet and the world-wide-web, embedded systems will increase the intelligence, control and communication capabilities of a wide range of objects, enabling their interaction and cooperation with people and organizations. Such smart objects will be joined together to create highly distributed systems, called Cyber Physical Systems, able to dynamically grow and adapt to the needs of individual users and communities. This convergence will bring a wealth of opportunities and innovations in technology, applications and business models. As a consequence, the potential to innovate of industries throughout the world will be influenced to a large extent by their capability to leverage the sophisticated technical solutions provided by the future embedded systems.

While the convergence between embedded systems and the Internet will open up huge opportunities for the ICT system sector in Europe, neither smart personal devices nor the cloud are among Europe’s strengths. Furthermore, the main actors in the Web are not European. Therefore, the competitive environment looks increasingly hostile. Europe will need to develop new technologies, applications and business approaches if it wants to remain at the forefront of ICT innovation and to address, through ICT, its societal challenges.
Recommendation 11.7: Embedded systems will increase the intelligence, control and communication capabilities of a wide range of objects, enabling their interaction and cooperation with people and organizations. Such smart objects will be joined together to create highly distributed systems to address for instance societal challenges. To support this evolution, R&D for Embedded Systems should be extended towards the system of systems area leading to Cyber Physical Systems. As a consequence, Europe should develop new scientific foundations, system design methodologies, development processes and tools to create the technical solutions tackling the challenges posed by system complexity (e.g. system behavior, dynamic growth, availability, fault tolerance, safety and security).

11.8 Human-Centred Computing and Multimedia

Although information technology touches every aspect of our lives, much software technology remains mired in a reductive “human-in-the-loop” paradigm. We need to shift our perspective, emphasizing a “computer-in-the-loop” approach that enhances human activity, placing users squarely at the centre of the design process. Human-Centred Computing combines the use of computers as tools that augment human capabilities (Human-Computer Interaction), as partners that provide intelligent support (Artificial Intelligence) and as a medium for human-to-human communication (Social Media).

Recommendation 11.8.1: Human-Centred Computing must integrate user-centred design methods into software design workflows with usability and attractive user experience, increase accessibility and integrate usable privacy primitives.

Within the next decade multimedia data will not only be linked to real world objects/situations and persons, but also multimedia data will be used as virtual representation of our real world. Real world objects and persons will be linked to their virtual counterparts documenting related information, development in time, functionality and user experience. All this will be realized in user-driven platforms, thus the borders between producers and consumers will blur.

Recommendation 11.8.2: Simulated Mixed Reality as enabling technology for a tighter coupling of reality and virtuality seeks for new solutions within the area of real-time numerical and optimization solution schemes, functional simulations, seamless rendering, analysis-through-synthesis and synthesis-through-analysis techniques. The overall vision should be to create perfect (virtual) assets being seamlessly integrated into reality.

11.9 Enterprise software

Enterprise software is a key driver of innovation in today’s business environment. The dynamic integration of business partners is of ever-growing importance in order to quickly react to
customers’ needs. A wide range of different services is increasingly offered and used in the Internet via cloud platforms in a cost-effective and efficient way. The new paradigm of “emergent software” is characterized by combining a wide range of components from different service providers in a dynamic and flexible manner, in order to fulfill the highly complex requirements of digital companies.

**Recommendation 11.9**: Develop a European strategic initiative on enterprise software technology to maintain Europe’s leadership. This initiative should stimulate to develop the next generation disruptive enterprise software technology and related worldwide standards. The world is facing the start of a new era of enterprise software given the ground breaking technologies such as cloud computing, many core, main memory database technology are all now available. In addition, the Internet of things has become reality generating a volume of data never envisioned in mankind history. Enterprise software needs to cope with this massive data to provide business process management taking real time real world information into account.

### 11.10 Future generation of software intensive systems

The key software technologies and research challenges stem from the specific characteristics of the software systems required for each domain in 2020. However, taking a broader perspective, we can identify several common qualities. A first key characteristic of the future generation of software-intensive systems in 2020 is the **massive number of nodes per system**, which will be one of the most visible features. A second area of interest is **adaptability**. In order to provide adaptability, system components must be able to understand one another, which is being complicated by the expected continued heterogeneous nature of future systems.

These future generations of software-intensive systems are called *ensembles*. Correspondingly, **ensemble engineering** is the science and engineering discipline of complex, integrated ensembles of computing elements. Ways to reliably and predictably model, design, and program ensembles are needed. Because of the social and economic importance of ensembles, the difficulty of building them, and the lack of engineering methods that adequately address the challenges posed by future software-intensive systems, we consider ensemble engineering to be one of the most important areas for European ICT research in the coming years.

**Recommendation 11.10**: Set up a FET Flagship to support the right timescales, levels of ambition and long-term funding that would allow Europe to maintain its pre-eminent position in the future generation of software intensive systems.
Authors and Contributors

ISTAG members

- Michel Cosnard – Chair (INRIA)
- Chris Hankin - Co-Chair (Imperial College London)
- Giovanni Barontini (Finmeccanica)
- Dieter Fellner (Fraunhofer Inst. for Computer Graphics Research, IGD)
- Wendy Hall (University of Southampton)
- Lutz Heuser (AGT International)
- Magnus Madfors (Ericsson)
- Vincent Marcatte (France Telecom)

Other contributors

- Serge Abiteboul
- Gérard Berry
- Tony Hoare
- Martin Wirzing
For further information
European Commission
Directorate General for Communications Networks, Content and Technology
Programme Coordination
CONNECT-F3