The TerraSwarm Research Center (TSRC)

A WHITE PAPER

David Blaauw, Prabal Dutta, Kevin Fu, Carlos Guestrin, Roozbeh Jafari, Doug Jones, John Kubiatowicz, Vijay Kumar, Edward A. Lee, Richard Murray, George Pappas, Jan Rabaey, Anthony Rowe, Alberto Sangiovanni-Vincentelli, Carl M Sechen, Sanjit A. Seshia, Tajana Simunic Rosing, Ben Taskar, John Wawrzynek, David Wessel

October 31, 2012

Lead Organization: University of California at Berkeley

Collaborating Universities: University of California at San Diego, California Institute of Technology, Carnegie Mellon University, University of Illinois at Urbana-Champaign, University of Michigan, University of Pennsylvania, University of Texas at Dallas, University of Washington.

Principal Investigator: Edward A. Lee, eal@eecs.berkeley.edu

The TerraSwarm Vision

Over the past decade there has been increasing interest in the use of "swarms" of sensors to help solve societal-scale problems. Sensor swarms, which can be wirelessly interconnected and deposit vast quantities of data in centralized repositories, offer an unprecedented ability to monitor and act on a range of evolving physical quantities.

Advances in design and manufacturing technologies have enabled the cost, size, power consumption, and variety of sensing and actuation devices and the associated networks to improve dramatically. Some industry observers predict that in ten years there will be thousands of smart sensing devices per person on the planet [9] (yielding a "tera-swarm"); if so, we will be immersed in a sea of networked real-world interface devices.

Sensor-based systems have already been proposed (and in some cases, deployed) for a broad range of monitoring (and even actuation) applications. But the potential goes far beyond what has been accomplished so far. When realized in full, these technologies can seamlessly integrate the "cyber" world (centered today in "the cloud") with our physical/biological world, effectively blurring the gap between the two. We call this emerging global cyber-physical network the "TerraSwarm," meaning that it encompasses many billions of sensors and actuators deployed across the earth.

Envision a futuristic "tale of two smart cities" with safe, efficient, and comfortable transportation and communication during the best of times, and secure, quick, and adaptable emergency response during the worst of times. Smart Cities use the TerraSwarm infrastructure to aggregate information from multiple sources, and use this information to (for example) automatically reroute traffic and identify health and safety threats, such as those created by an earthquake or a terrorist attack. TerraSwarm applications identify individuals who can benefit from information that has been gathered, and notify them using local resources such as cell phones, nearby displays, or audio systems. These systems can be used to form response teams and implement a range of rescue and security operations.

TerraSwarm applications are characterized by their ability to *dynamically recruit* resources such as sensors and data from the cloud, aggregate and use that information to make or aid decisions, and then dynamically recruit actuation resources — mediating their response by policy, security, and privacy concerns.

Achieving this vision will require a three-level model. The **cloud backbone** will offer extraordinary computing and networking capability, along with global data analytics, access, and archiving. Mobile battery-powered **personal devices** with advanced capabilities will connect opportunistically to the cloud and to nearby **swarm devices**, which will sense and actuate in the physical world.

Ubiquitous connectivity between the cloud and mobile devices such as smartphones is already (nearly) a reality, and through common and general programming and communication interfaces (e.g., "app" programming and TCP/IP) this connectivity has turned the cloud+mobile universe into a flexible platform enabling millions of applications that we could not have imagined a few short years ago. These parts of the system will continue to develop rapidly under large-scale commercial investment. The swarm level, however, because it directly interacts with the physical world, presents challenges that demand forward-looking research. The potential payoff of such research is a system that can fundamentally change and empower human interaction with the world.

Current “smart” applications, such as smart homes, smart grids, and battlefield management systems, typically address a single application on a dedicated set of resources. While this approach provides performance guarantees and reliability, it prevents economies of scale, and, more importantly, it prevents the explosion of possibilities that results from sharing data and devices across applications. The TerraSwarm vision cannot be achieved by a single vendor providing the components as an integrated system. What is needed instead is the swarm equivalent of the common, general, “app” framework that has recently enabled smartphones and similar devices to rapidly deploy and serve a vast range of often unanticipated applications by recruiting resources and composing services. The swarm will never achieve its potential without a “SwarmOS” on which such “swarm-apps” can be built and composed by millions of creative inventors.

While open architectures with dynamically recruitable resources can open up significant security and privacy risks, they can also make systems more efficient (through sharing of resources), more resilient (through dynamic reconfiguration leveraging redundant resources), and more capable, enabling applications we have not yet invented or that cannot yet be realized. Reliability, robustness, adaptability, and security must be built in from the start.

CENTER MISSION: The TerraSwarm Research Center (**TSRC**) aims to enable the simple, reliable, and secure deployment of advanced distributed sense-control-actuate applications on shared, massively distributed, heterogeneous, and mostly uncoordinated swarm platforms through an open and universal systems architecture.

When the World Wide Web was first launched, few people would have predicted the astounding range of applications that it would enable. It has profoundly changed the way people interact and behave, how businesses are run, and how information is exchanged. We believe that swarm-based systems will ultimately dwarf the impact of the Web, and that it is essential to provide a collaborative environment in which to address TerraSwarm’s extraordinarily wide range of challenges and opportunities. By viewing key challenges through many different eyes, we expect to be able to generate more innovative ideas and solutions.

The TerraSwarm Challenge

The TerraSwarm vision holds enormous promise, but poses a number of daunting challenges. These include the following:

- Swarm systems rely on vast numbers of heterogeneous sensors that are generating massive amounts of data. How will this data be stored, accessed, processed, and interpreted?
- If the data is used for security- or safety-critical systems, how can we verify that it is accurate (i.e., that the sensors are functioning properly) and that it has not been compromised (i.e., that is secure from malicious or even inadvertent tampering)?

- TerraSwarm applications are generally cyber-physical systems that involve physical actuation, and hence will have stringent testing and verification requirements. But they will also be highly dynamic, adapting their structure and recruiting resources on the fly. How can testing and verification extend to continuously evolving systems? How can we ensure that effects on the physical world are safe?
- How will new applications be conceived and developed? How will they change after deployment as they continuously reconfigure and adapt? What innovative applications are within the realm of possibility, and how shall we prioritize their development?
- How can we develop and evolve applications and systems in a manner that is both cost-effective and energy-effective?
- How will we address data privacy, security, and safety?

Nearly every science and engineering university in the country is engaged in research that is either directly or peripherally applicable to swarm systems and could potentially be used to address these challenges. Relevant research areas include sensor technologies, actuators, semiconductors, communication systems, control systems, robotics, data analysis, data mining, modeling and simulation tools, operating systems, energy efficiency technologies, machine learning, data security and encoding, and cyber-physical systems, among others. Thus far, there has not been a coherent effort to bring together these disparate research efforts to serve swarm-based application development – yet the swarm will only reach its full potential when it becomes a unified, standardized platform enabling the development of "swarm apps."

To this end, the TerraSwarm Research Center will provide a home base for researchers to coordinate efforts and exchange ideas, potentially leading to much more rapid and efficient deployment of TerraSwarm applications.

TerraSwarm Center Research Themes

The research agenda of the TerraSwarm Research Center is broad and ambitious, spanning components, architectures, services and functions, security and privacy, methodologies, algorithms and tools, and design flows. To inspire the overall effort, the team has chosen a "Smart City" concept as the integrating theme involving every center member. Technology development will be structured around three additional themes: Platform Architectures and Operating Systems; Services, Applications and Cloud Interaction; and Methodologies, Models, and Tools.

The four research themes are discussed in more detail below.

Theme 1: Smart Cities

The TerraSwarm Research Center will focus on a single integrated challenge problem: the application of TerraSwarm technologies to Smart Cities. Two scenarios are of interest: a city during normal operation and a city during natural or man-made disasters (such as accidents, failures, earthquakes, or terrorist attacks).

In normal operation, a swarm-enabled “Smart City” not only helps run the infrastructure more effectively but empowers its occupants by providing more effective interfaces, better mobility, and experiences in immersive realities in a way not possible before. For example, maintenance crews may recruit sensors from underground utilities, and combine that sensor data with data from pipe-crawling robots and from the cloud. They can use this information to guide maintenance operations using overlay displays in a manner similar to what televised sporting events use, based on contextual 3D information.

In emergency scenarios, the swarm-enabled Smart City is able to safely and securely align both stationary (e.g. biohazard detection sensors) and mobile (e.g. UAVs and robots) resources needed to protect itself and its inhabitants. Depending on the scenario, it may be impractical to rely on human operators to remotely pilot vehicles, so the mobile network must be able to autonomously deploy itself in a region of interest. Environmental sensors will focus on detecting and alerting inhabitants of dangerous chemical and biohazards, while immersive environments created on the fly can enable teams to deploy clean-up and security forces. The goal of the network under emergency conditions is to adapt, coordinate, respond, and resolve dangers appearing in the environment effectively, efficiently, and as autonomously as possible.

In both scenarios, Smart Cities combine the management of fixed infrastructure (e.g. environmental monitoring, energy-usage, tracking and mapping), mobile assets (automatic vehicles, UAVs, robots), and immersive humans in an integrated whole. This involves seamless discovery and integration of sensing, actuation, and computation, with the use of feedback to manage uncertainty.

Building on testbeds in the constituent universities, we plan to realize a number of Smart City scenarios integrating multifunctionality, dynamic adaptation, safety and security, scalability, and robustness. The critical research issues to be addressed include how to recruit and compose heterogeneous resources, how to dynamically adapt applications to changing resources and contention for resources, and how to share resources without compromising safety, security, or privacy.

Theme 2: Platform Architectures and Operating Systems

In a TerraSwarm system, applications will compete for a variety of resources, including sensors, actuators, networks, computing resources, storage, energy, and wireless spectrum. The goal of this theme is to develop architectures and operating systems that can dynamically balance the competing needs of distributed concurrent applications so that functionality, robustness, utility, and quality of service are guaranteed. The systems support for this adaptive, resource-aware vision is the SwarmOS, a highly distributed infrastructure that touches every node in the system. Its purpose is to efficiently allocate resources based on complex optimization strategies, while maintaining appropriate security and privacy.

The SwarmOS must support continual reconfiguration of applications and of its own service definitions without ever having the luxury of a clean restart. It must also support richly heterogeneous components including sensors, actuators, networks, and computers, and it must tolerate appearance or disappearance of resources. It must be distributed and mobile, orchestrating actions across heterogeneous networks.

Central to our approach is a methodology that decomposes applications into interconnected graphs of services, borrowing important ideas from service-oriented architecture (SOA) such as loose-coupling, service abstraction, discoverability, and composability, while avoiding much of the overhead and baggage of SOAs. We will supplement the service interface with utility guarantees, provided as service level agreements or contracts.

A key goal of the TerraSwarm project is to define the abstractions for services and locations so that the resource management infrastructure can compose resources (such as sensors and actuators) adaptively. To achieve this goal, we require simple and energy-efficient discovery; built-in security from the ground up; guarantees in the form of temporal and limited-duration service-level agreements (SLAs) based on self-assessment of internal and external conditions, availability of resources, workload, and required quality of service; and well-defined lifetimes for which the above are valid.

The security of information, actuation, and brokerage is essential to the success of the TerraSwarm vision. For the leaf nodes of the TerraSwarm system (the sensors and actuators), it is important that security mechanisms are built-in, yet do not constitute an energy burden. This leads to a need to develop *energy-efficient* hardware support for encryption/decryption as well as hardware-enforced key management.

Theme 3: Services, Applications, and Cloud Interaction

The TerraSwarm vision is one of composable services that can be dynamically recruited by applications. Formally, applications are defined as dynamic, distributed graphs of connected services. Both "dynamic" and "distributed" are important here; applications persist even as the individual components that comprise these applications change. This view elevates the concept of an *integrated modular architecture* (**IMA**), today's target for systems-of-systems design, from the system level to the enterprise level, and augments it with discovery and run-time adaptation.

Control as a Service. From the user perspective, the TerraSwarm provides (contextual) total awareness, which is enabled by a dynamically changing mixture of local and remote swarm sensors. Adaptive services will exploit these devices to improve accuracy and quality for the user. Ensuring that such adaptive services remain effective, efficient, and safe under dynamic restructuring is a challenging control problem. The TerraSwarm vision is to decentralize the design of such systems, improving their robustness and making them more adaptable and opportunistic. Control strategies will be synthesized on the fly from goal specifications and constraints, a vision we call **control as a service**.

The Cloud as a Companion. A central challenge to be overcome is the imbalance between the massive amounts of information that could be collected and the time-sensitive interests and needs of the user(s). A naïve approach is to collect and store all data, and have cloud-based services distill the information for user consumption. But the most interesting services will need the right (contextual) data at the right time and the right place. Closed-loop cyber-physical interactions will not tolerate the latencies incurred by cloud-based archiving and indexing. Moreover, the vast data flood that will emerge from the

Terraswarm make this naïve approach far too costly, even with huge advances in storage technology.

In the TerraSwarm environment, resources are recruited opportunistically based on availability and need, with the objective of providing the best possible experience to the user. Data produced locally will be maximally leveraged locally. Nevertheless, the cloud plays an essential role. A key goal of this theme is to "wake up" the cloud, giving it a physical rather than just cognitive presence; rather than just providing information, a TerraSwarm system will affect our physical environment.

As with social networks and information search technologies, the cloud participates by aggregating data from a multiplicity of sources, something not possible on a single physical device, no matter how much computation and memory capability it has. The cloud is not just a computation and memory resource; it is an information aggregator and a service synthesizer. Data aggregation allows us to shift feedback control from the system level to the enterprise level.

<u>Structured Data Summarization.</u> The vast quantity and variety of swarm data will require new approaches for correlating, interpreting, and displaying data in a meaningful way. We plan to develop effective mechanisms for managing swarm data.

<u>Secure and Safe Swarms.</u> The web and social media have opened the floodgates of personal information available about us even to strangers. Even as our culture is only starting to learn to deal with the consequences of that information flood, that flood is about to be itself overwhelmed by data streams from physical sensors. The TerraSwarm vision is that security and privacy must be built into the very core of service definitions.

The TerraSwarm project will use a system theoretic formulation to address privacy concerns, defining filters that release useful information without compromising privacy. Our proposed approach relies on the notion of differential privacy [2], which provides strong privacy guarantees against adversaries with arbitrary side information.

We will also examine potential data leakage introduced by composable services through side channels such as timing and power consumption. Fortunately, there are synergies. For example, temporal isolation may be introduced to guarantee resources to safety-critical services, but it can also be used to prevent side-channel attacks, where private information is deduced from temporal variations in software execution.

We will explore the use of security-related technologies and techniques such as static analysis, hazard analysis, and elliptic curve cryptography to implement effective security approaches. We will leverage existing research in the area of distributed storage [5, 7, 6, 3, 1] to inform the design of cloud-based swarm applications that need strong guarantees of security despite their reliance on physically insecure infrastructure.

Theme 4: Methodologies, Models, and Tools

A key challenge in designing TerraSwarm applications and infrastructure is that the distinction between "design time" and "run time" becomes blurred. Ensuring that different components and subsystems can be dynamically recombined yet still function properly will require new, highly advanced development methodologies, models, and tools. Functions to be realized must be separated from the components that will be used to realize them (the "separation of concerns" concept – see [8]). Programming models must be less centered on algorithms (step-by-step transformation of data) and more centered on dynamics (change of state over time), distribution, discovery, and adaptation. Optimizations that might be performed at design time in a conventional system-of-systems, such as mapping of functions to resources, will need to be performed at run time. Design-time testing and verification will not be adequate, because components and applications are dynamically composed and recomposed. Validation will need to be performed at a higher level, will need to cover families of possible run-time configurations rather than just one, and will need to include run-time validation strategies that are lightweight and energy efficient.

The goal of this theme is to develop methodologies, models, and tools that support the unique requirements of TerraSwarm systems. Critical research areas include advanced modeling, verification, and adaptation approaches, described further below. This theme is intertwined with the others and provides the theory and tools support to the entire TerraSwarm effort.

Modeling. Developing TerraSwarm systems will require the ability to effectively model system components and their interactions. Models must capture the evolving availability of services and resources, which can potentially be combined to provide many different types of applications. Models must also capture the rules for recruiting and combining resources and services. Current modeling approaches do not support the complex, dynamically changing characteristics of TerraSwarm systems.

We will model TerraSwarm systems as a dynamic hierarchical graph of components that comprise the system. The nodes of the graph will represent services. Since these graphs are hierarchical, a node may itself be a graph aggregating sub-services to define a new service. The edges in the graph represent (i) communication paths between components; (ii) authority relations between components; (iii) use relationships (i.e., service x uses service y); (iv) ownership relations; (v) coordination; (vi) controllability; and (vii) observability.

A TerraSwarm system's behavior is the result of run-time and design-time optimization processes that choose the configuration that will best achieve its goals. A **configuration** of a TerraSwarm system is a particular graph structure that selects specific capabilities of the nodes in the graph (i.e., subsystems).

Verification. Verification of TerraSwarm systems' functionality will be difficult. The large number of components, their heterogeneity, and the dynamically changing structure will render exhaustive formal verification impractical. Instead, we will need compositional and incremental techniques. Compositional techniques hierarchically infer properties of compositions from properties of components. Incremental techniques infer properties of a configuration from properties of a similar configuration.

Compositional verification is enabled by assume-guarantee reasoning, which requires models of the environment. (Assume-guarantee contracts are described in [4].) In a dynamic TerraSwarm context, these models will likely be incomplete, and hence will need to be inferred or refined from observations. Such models will be imperfect, and therefore should include metrics of uncertainty that verification techniques can reason about.

In the dynamic network of a TerraSwarm system, non-interference properties will become key. For example, when a node joins or leaves a network, it must not disrupt any service that does not depend on this node. Non-interference of temporal properties becomes particularly important for closed-loop cyber-physical systems, because if one service disrupts the timing of another, it may change the dynamics of a physical system in undesirable ways. Hence, models will need to include temporal specifications that verification techniques can reason about.

Finally, not all nodes will be equally trusted. TerraSwarm protocols will need to detect compromises, distinguished trusted from untrusted data and resources, and be robust to the presence of a certain number of malicious nodes. Techniques based on a combination of formal methods and algorithmic game theory can be effective in analyzing the impact of malicious agents. Tools for formal verification of composition and abstraction-refinement relations using horizontal and vertical contracts will be developed together with other verification techniques based on functional and architectural simulation. Indeed, contracts are critical in defining and verifying compositionality and the abstraction-refinement relation among the different layers of abstraction of a TerraSwarm system.

Adaptation. TerraSwarm applications will need to deploy resources dynamically in order to achieve mission goals, and these goals may change based on circumstances encountered in the field. Typical optimization strategies for determining how best to deploy resources depend on knowing the spatial probability distribution of relevant events — but in a TerraSwarm system, this distribution will not be known in advance.

By leveraging theoretical and algorithmic tools developed for adaptive systems, we can derive new simple algorithms for complex tasks, such as coverage, source seeking, distributed partitioning, and tracking under uncertain communication constraints. These algorithms do not depend on a model of the environment, exploiting instead event observations during deployment. Moreover, they adapt to slowly varying environmental conditions or sudden but infrequent environmental changes.

Initial TerraSwarm Test Applications

A key characteristic of the Smart Cities infrastructure, and of the TerraSwarms approach in general, is that the infrastructure is shared among multiple applications. In the TerraSwarm project, we will be developing a general-purpose infrastructure whose characteristics will be tested on a few well-chosen applications.

Initially, we will focus on the following application classes, which may be adapted as the project progresses.

- *Consumer Applications.* TerraSwarm systems enable a much richer set of consumer applications because of their interactions with the physical world. Consider, for example, a *smart jukebox*, which is a relatively simple application that incorporates several key TerraSwarm characteristics. During normal city operation, it uses information about local demographics and listening preferences to generate a customized playlist, which can then be used by restaurants (or other public meeting spaces) to adapt their soundscapes to the preferences of their customers on a dynamic basis. Leveraging the work at the Berkeley CNMAT (Center for New Music and Audio Technology), it is even possible to deliver different soundscapes to different locations within a public forum (using beamforming and very large speaker arrays), and to extend to soundscape synthesis rather than just delivery. Interaction devices such as touchscreen tables could extend the smart jukebox into the social networking world, allowing for participatory soundscapes that go well beyond Karaoke.

The smart jukebox will require semantic localization, analysis of personal information available from mobile devices and social networking databases, and dynamic resource recruiting and control. The application will be required to construct models of musical preferences, infer models from sample behaviors, define optimization criteria, construct statistical models of user populations and system dynamics, analyze the system dynamics, identify optimization algorithms, analyze privacy and security, and optimize the delivery mechanism according to the available resources. It can leverage existing machine learning technology used in (for example) Pandora and Apple iTunes' Genius Bar, both of which aggregate information about musical preferences and make predictions about new songs that are likely to be enjoyed.

In emergency scenarios, the smart jukebox infrastructure can be used to identify the location of people with relevant skills (e.g., doctors, electricians, off-duty police officers) and alert them via the localized sound system or text message that their skills are needed at a nearby location. By aggregating information about available human resources and their locations, the system can more effectively direct resources to appropriate locations and optimize emergency response times.

Although its utility in normal, day-to-day operation is not critical, the smart jukebox is a new, technologically challenging application that will serve as a good test case for key aspects of the TerraSwarm tools and methodologies.

- *Autonomous Vehicle Response.* At least one of our test applications will involve recruiting and deploying autonomous vehicles. These may include, for example, cars, aerial drones, or micro-robots, which may be required to operate alone or within coordinated groups. The range of possible uses for autonomous vehicles is huge. For example, in the best of times, they can be used for accident and crime prevention; in the worst of times they may be used for emergency response, rescue efforts, surveillance, or delivery of medications. This test application will leverage ongoing work at the University of Pennsylvania and Caltech in the design of vehicle trajectories, control laws, and decision-making protocols for autonomous vehicles, including micro UAVs (unmanned aerial vehicles). Tasks that must be performed by these vehicles include collecting information using mobile sensors, transporting physical objects and/or people, establishing and maintaining *impromptu* communication — all of which must coexist with other (human-operated) vehicles.

  Under emergency conditions, mobile vehicles must be capable of operating as individual units, in ad hoc groups established by local proximity, or as a city-wide resource, with intermittent communication capability. Real-time, distributed algorithms for ag-

Name	Institution
David Blaauw	Michigan
Prabal Dutta	Michigan
Kevin Fu	Michigan
Carlos Guestrin	U. of Washington
Roozbeh Jafari	UT Dallas
Doug Jones	UIUC
John Kubiatowicz	Berkeley
Vijay Kumar	Penn
Edward A. Lee	Berkeley
Richard Murray	Caltech
George Pappas	Penn
Jan Rabaey	Berkeley
Anthony Rowe	CMU
Alberto Sangiovanni-Vincentelli	Berkeley
Carl M Sechen	UT Dallas
Sanjit A. Seshia	Berkeley
Tajana Simunic Rosing	UCSD
Ben Taskar	U. of Washington
John Wawrzynek	Berkeley
David Wessel	Berkeley

Table 1: Key Academic Personnel.

gregation of information, interaction with cloud services, and cooperative control and decision-making will be tested in this context and used to explore new TerraSwarm services and applications.

- *Health-Related Applications.* The TerraSwarm infrastructure (together with the cloud) will have access to a variety of health- and lifestyle-related data, including people's location, activity, and vital signs (via mobile devices and wearable sensors, as well as imagers embedded in the surrounding environment); environmental conditions (via networked sensors); and social connections (via the social networking infrastructure). Some of this information may be provided by streams of data from innovative sensors, such as energy-harvesting wearable sensors, or from wall-size imagers. To close the loop, analysis of data from such sensor streams might be used to guide people towards healthy activities or to optimize the performance of troops, police, and medical personnel.

Key Academic Personnel and Facilities

Key academic personnel for the TerraSwarm Research Center are listed in Table 1.

The TerraSwarm Research Center has assembled a world-class team whose expertise spans the diverse disciplines required by this project. The team includes experts in large-scale, adaptive, cyber-physical control systems (Kumar, Murray, and Pappas); programming models and tools for heterogeneous, real-time, and distributed cyber-physical systems (Lee, Sangiovanni-Vincentelli, and Seshia); security in systems with dynamic topologies (Fu and Kubiatowicz); machine learning (Guestrin and Taskar); privacy (Pappas);

networked sensor and actuator platform design (Blaauw, Dutta, Jafari, and Sechen); signal analytics (Jones and Wessel); wireless networking and distributed systems (Rabaey); system architecture (Kubiatowicz, Rowe, and Rosing); and application platforms (Kumar, Rosing, Wawrzynek, and Wessel). Each of these contributors has made significant, innovative contributions to their areas of research.

The TerraSwarm Research Center will be headquartered in the Electrical Engineering and Computer Sciences (EECS) Department at UC Berkeley.

References

[1] P. Druschel and A. Rowstron. Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility. In Proc. of ACM SOSP, 2001.

[2] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, Theory of Cryptography, volume 3876 of Lecture Notes in Computer Science, pages 265–284. Springer Berlin / Heidelberg, 2006. Available from: http://dx.doi.org/10.1007/11681878_14.

[3] K. Fu, M. F. Kaashoek, and D. Mazières. Fast and secure distributed read-only file fystem. In Proc. of USENIX Symp. on OSDI, 2000.

[4] C. B. Jones. Specification and design of (parallel) programs. In R. E. A. Mason, editor, IFIP 9th World Congress, Information Processing, pages 321–332. North-Holland, 1983.

[5] J. Kubiatowicz et al. Oceanstore: An architecture for global-scale persistent storage. In Proc. of ASPLOS, 2000.

[6] D. Mazières, M. Kaminsky, F. Kaashoek, and E. Witchel. Separating key management from file system security. In Proc. of ACM SOSP, 1999.

[7] S. Rhea, P. Eaton, D. Geels, H. Weatherspoon, B. Zhao, and J. Kubiatowicz. Pond: the OceanStore prototype. In Proc. of USENIX FAST, 2003.

[8] Alberto Sangiovanni-Vincentelli. Quo vadis SLD: Reasoning about trends and challenges of system-level design. Proceedings of the IEEE, 3:467–506, March 2007.

[9] Mikko A. Uusitalo. Global vision for the future wireless world from the WWRF. IEEE Vehicular Technology Magazine, 1(2):4–8, 2006.