A high-confidence system is one in which the designers, implementers, and users have a high degree of assurance that the system will not fail or misbehave due to errors in the system, faults in the environment, or hostile attempts to compromise the system. Consequences of such systems’ behavior are well understood and are predictable under an operational context envisioned by its creators. High-confidence systems (HCS) are highly secure and robust: they can withstand various threats, malicious attacks, and hardware/software component failures. Correctness, predictability, reliability, availability, security, and survivability are the key properties that constitute the basis of high confidence.

As computer systems and computer networks are being used to store, transmit, and process secret, highly valuable information, to generate products and services for human consumption, and to control mission critical systems, the design and deployment of high confidence distributed systems has become a crucial issue. Protection of hardware and software infrastructures is critical to the privacy of citizens, the safety of transportation systems, financial health of business organizations, stability of the global economy, and national security. Last year, a blizzard of attacks on the Internet and E-Commerce web sites (like Yahoo, CNN.com, eBay, amazon.com, etc.) revealed that the software and information base that these companies employ are highly vulnerable not only to security penetration, but also to denial of service through various forms of security attacks: disclosure, malicious modification, deletion, viruses. Privacy of individuals is compromised through inappropriate access to information. In a recent incident, a hacker gained access to the names, diagnoses, home addresses, and social security numbers of hundreds of patients at a large public university medical center. A large number of other break-ins go undetected or unreported. The case is similar for safety. We are relinquishing control to rapidly increasing automation in our homes, automobiles, and environments. The safety consequences of failure range from inconvenience to life-threatening. For example, in automobiles as braking, ignition, steering, transmission and other automotive functions become increasingly automated, complex, and dependent on software and information technology, the number of software-related incidents can be expected to rise.

Design, development, validation, and testing of secure, safe software for networked systems are central research problems. In the past, software loopholes have contributed to a majority of security breaches and to catastrophic aviation and automation failures such as the Ariane 5, the Mars Probe, and Airbus accidents. In the past, development of software has been ad hoc. As our dependence on information technology increases, there is a need to develop a sound, theoretical basis for building trustworthy software systems that meet safety and security requirements. Digital cash, secure execution of untrusted (mobile) code, and secure, dependable mobile computing systems are important emerging problems. We expect to have confidence in networked embedded software applications in our homes, workplaces, and vehicles. We need to understand well the implications of software design and structuring for vulnerability in these systems to failures, extreme environmental events, and security attacks. This will require both fundamental and empirical research to establish, test, validate, and improve secure networked and software-enabled system construction principles. There is a need for formal design
methods, programming technology, and system software services for increased security and for validating properties of safety-critical software. We need measures and technology to assess the levels of safety or security actually achieved by a system. Certification effort is costly and can impede adoption of new software-enabled technologies. We need alternatives to current testing-based certification practice. New monitoring technology (e.g., high-speed packet filtering mechanisms) is needed that will work effectively with the ultra high-speed networks of the future, as well as forensic tools to analyze the nature and causes of accidents and intrusions.

Research is needed on critical hardware and software technologies that are necessary to achieve high level of system safety, security, reliability, privacy, and survivability. This requires expertise from multiple disciplines like operating systems, software engineering, architectures, theory of computing, and networked systems. Research is needed to meet the following key technical goals towards high confidence systems: (i) to develop a sound, theoretical basis for assured construction of safe, secure systems. (ii) to develop principles and methodology for secure and reliable software design and development. (iii) to develop techniques for verification and validation of high confidence software against security breaches and hardware/software faults. (iv) to develop tools and methodologies for the design, construction, and evaluation of behavioral enforcement mechanisms. (v) to develop mechanisms to reduce the effort, time, and cost of information systems design, development, and analysis while simultaneously increasing the level of confidence that can be achieved. (vi) to furnish reference implementations that embody innovative HCS methods and techniques to enable rapid adoption. (vii) to develop measures of the performance and measures of effectiveness to quantify the improvement in the system-level confidence that can be achieved through developed HCS technologies.

To implement these objectives, research is needed in the following four categories: (1) HCS foundations: research here will develop the supporting theory and scientific base for high confidence systems. (2) HCS tools and techniques: research will construct means to apply the foundations of HCS to design and build large-scale HCS systems. (3) HCS engineering and experimentation: this research will provide reference implementations, proof-of-concepts, reusable tools and techniques and empirical evidence of HCS capabilities and limitations. (4) pilot projects: This will demonstrate how to apply HCS technologies to specific user domains.