The Evolution of Dependable Computing at the University of Illinois

Dedicated to Professor Algirdas Avizienis

R. Iyer, W. Sanders, J. Patel, Z. Kalbarczyk
Center for Reliable and High-Performance Computing
Department of Electrical and Computer Engineering and
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign

Long Association with FTCS (now DSN)

- Al Aviziennis
  - Ph.D. Illinois, Founder FTCS and IFIP WG 10.4
- Program Chair or General Chair
  - Gary Metze, faculty, FTCS-2, FTCS-4
  - John Hayes, Ph.D. Illinois, FTCS-7
  - Jacob Abraham, faculty, FTCS-11, FTCS-19, IEEE TC on Fault-Tolerant Computing chair
  - Ravi Iyer, faculty, FTCS-19, FTCS-25, IEEE TC on Fault-Tolerant Computing chair
  - Kent Fuchs, faculty, FTCS-27 IEEE TC on Fault-Tolerant Computing chair
  - Ram Chillarege, Ph.D. Illinois, FTCS-28
  - Bill Sanders, faculty, FTCS-29 IEEE TC on Fault-Tolerant Computing chair
  - Zbigniew Kalbarczyk, Ph.D., DSN-02
  - Tim Tsai, Ph.D. Illinois, DSN-04

Early Developments – Illiac-I and Illiac-II

- During 1950's and 1960's Illiac-I and Illiac-II
  - Frequent failures of vacuum tubes jump started the early work in fault diagnosis
  - A subtle design bug in the arithmetic unit, *(-2) * (-2) giving (-4),* escaped the tests
    - caught after about nine months by a numerical double-check built into a user's program
  - While no attempt was made to model faults systematically, the handshake mechanism used in the ALU control exhibited the basic idea of what is now called self-checking operation.
  - Avizienis conducts early research on parallel computer arithmetic

Pioneering Work of Seshu and Metze in Test and Diagnosis

- Sequential Analyzer (Seshu), which included a set of programs for
  - automatic generation of fault simulation data (stuck-line faults) for a given logic circuit and test sequence
  - automatic generation of test sequences for combinational and sequential circuits.
  - studying computer self-diagnosis
- Sequential analyzer was applied in at Bell Telephone Laboratories to improve diagnosis procedures for ESS-1
- Chang, Manning, and Metze produced, as a tribute to Seshu; probably the first book devoted entirely to digital fault diagnosis.
- PMC (Preparata, Metze, Chien) model for computer self-diagnosis

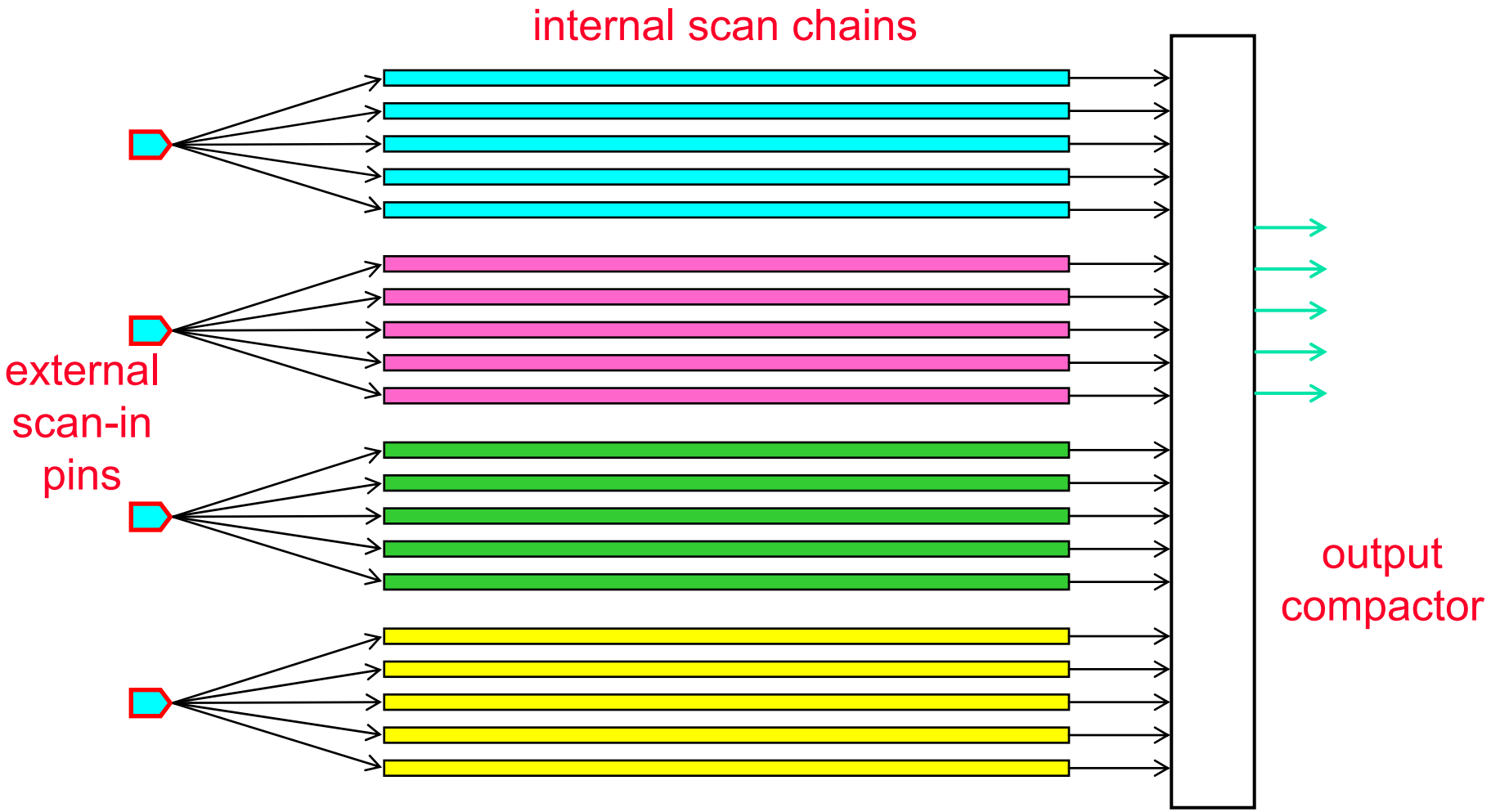
Automatic Test Pattern Generation (ATPG) – from Research to Product

- First commercial ATPG in 70's (Marlett) and then several updated versions through 80's and 90's
- Founding ATPG company *Sunrise Test Systems,* now a part of Synopsis (Niermann and Patel)
  - Commercialization of PROOFS, the fastest, most memory efficient fault simulation algorithm, in 1990
  - PROOFS now used in all commercial ATPG tools

Testing of multi-Million gates – Scan Design

- Early Scan Design
  - NEC, IBM and William-Angel of Stanford, all claim to be the first to propose Scan
  - Others claim that the first proposal of the scan idea was in the book by Chang, Manning and Metze of Illinois!
- Today large circuits with close to a million flip-flops in scan chain requires enormous test time and data
- Illinois Scan Architecture (1999-2002)
  - Reduces test time and test data by factors of 100 or more with no logic overhead!
  - Already in use at IBM, Intel, Syntest and others

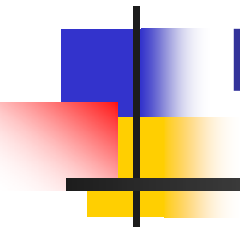
Illinois Scan Architecture



Functional-Level Test Generation

- Memory testing using functional-level fault model (i.e., without the availability of information about their internal structure)
  - the initial fault model for memories included stuck bits in the memory and coupling between cells in a memory.

- Test generation procedures for microprocessors based on an extrapolation of the approach to testing memories
  - A general graph-theoretic model at the register-transfer level to model microprocessors using only information about its instruction set and the functions performed.
  - A fault simulation study on a real microprocessor showed extremely good fault coverage for tests developed using these procedures.



Detection & Recovery

Self-Checking and Time Redundancy

- Pioneering work of Carter led Anderson and Metze (at Illinois) to formulate Totally Self Checking (TSC) circuits; subsequent work led to introduction of Strongly Fault Secure property
- Triggered widespread research in academia and industry in the 70's and 80's
- Metze proposed time redundancy techniques for checking errors based on Alternating Logic; subsequently enhanced as RESO (Recomputing with Shifted Operands)

Algorithm-based Fault Tolerance – ABFT and Checkpointing

ABFT

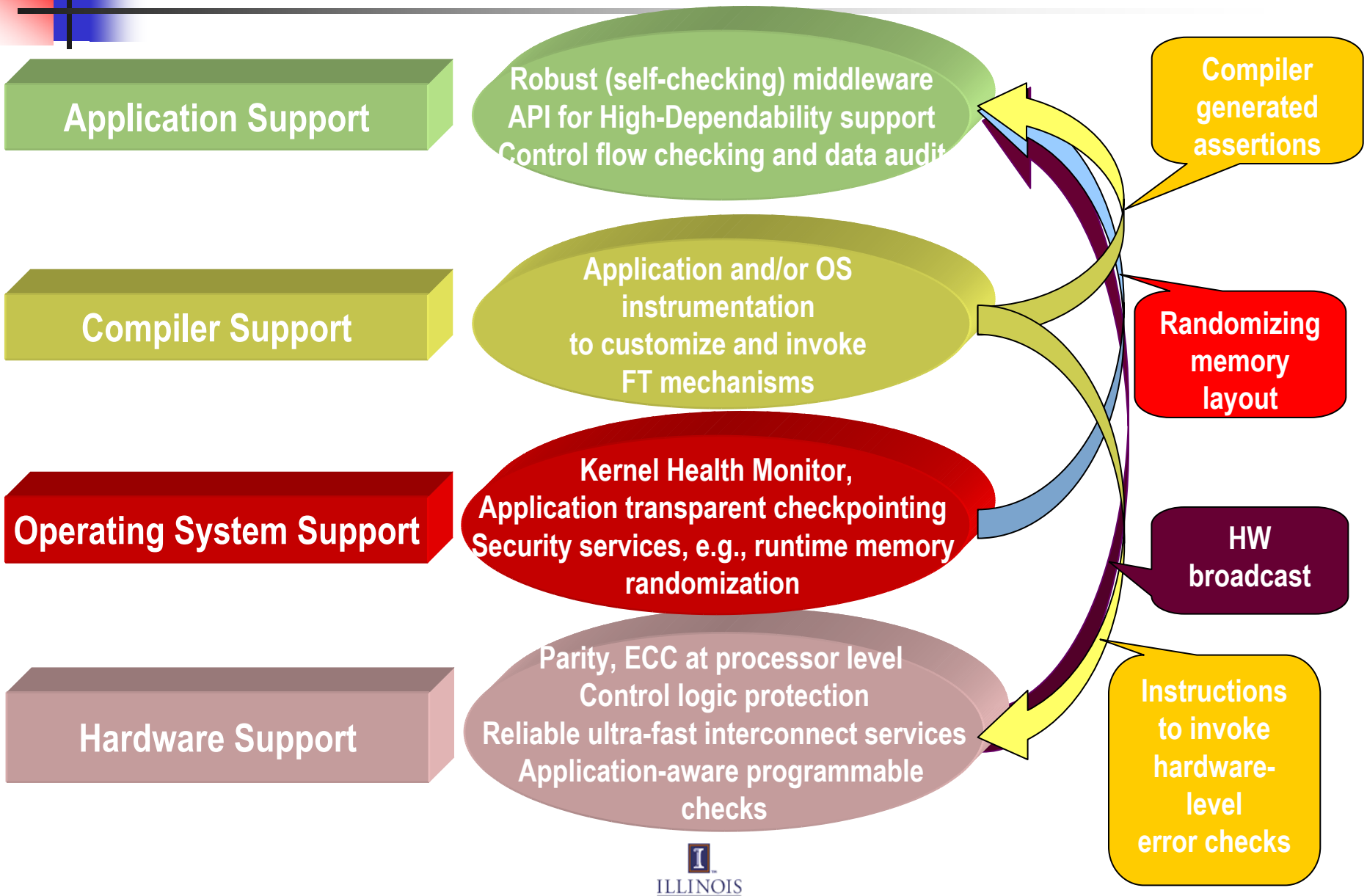
- Matrix encoding schemes for detecting and correcting errors when matrix operations are performed by processor arrays
- Generalized to linear arrays, Laplace equation solvers, and FFT networks
- Ideal for low-cost fault tolerance for special-purpose computations, including signal-processing applications
- Explored by a large number of researchers,
  - more recently as part of the REE program at JPL

Checkpointing

- Asynchronous checkpointing for distributed systems optimized for space overhead and performance
- Compiler-assisted multiple instructions rollback scheme to aid in speculative execution repair in microprocessors
- Low-overhead coordinated checkpointing for long-running parallel and high-availability applications
- *RENEW* toolset for rapid development and testing of checkpoint protocols

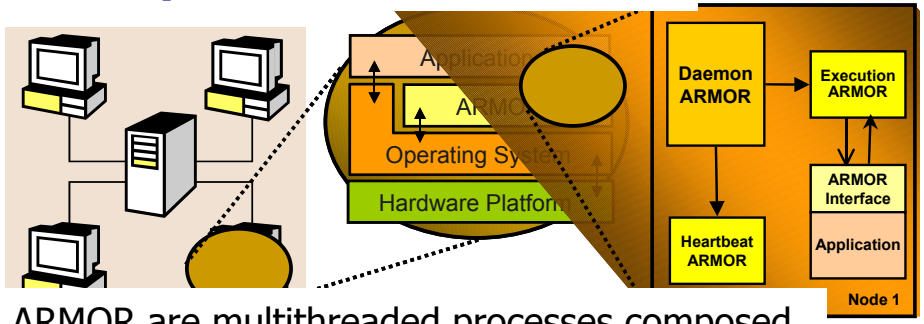
Current Directions

Hierarchical Application Aware Error Detection and Recovery



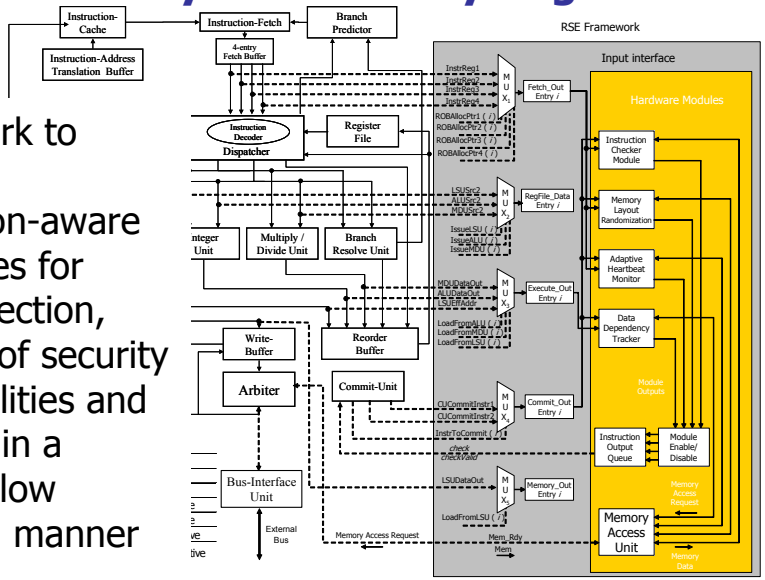
Middleware and Hardware Frameworks for Fault Tolerance and Security

ARMOR – High Availability and Security Middleware



ARMOR are multithreaded processes composed of replaceable building blocks – elements, which implement error detection, recovery policies, security services, and management of runtime environment.

RSE – Reliability and Security Engine

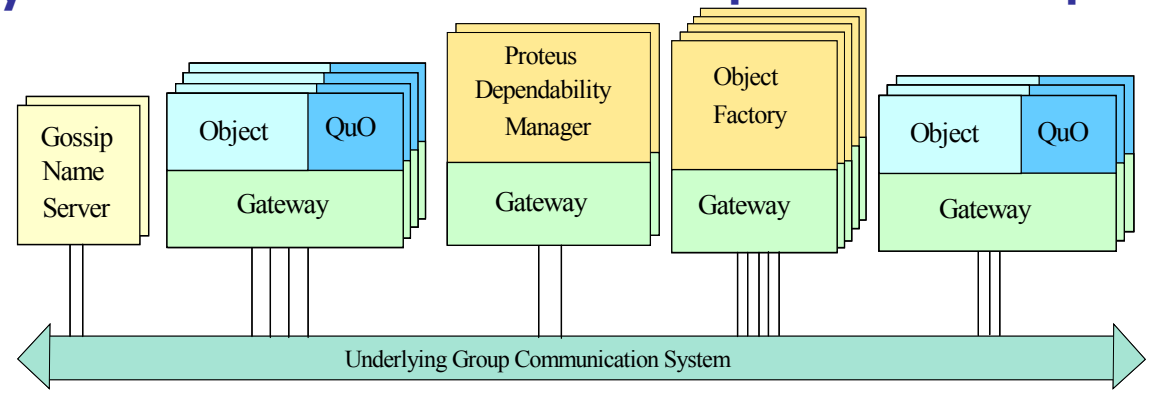


Framework to provide application-aware techniques for error-detection, masking of security vulnerabilities and recovery in a uniform, low overhead manner

AQuA – Adaptive Quality of Service for Availability

Algorithms and architectures for building dependable, object-oriented, distributed computer/communication systems

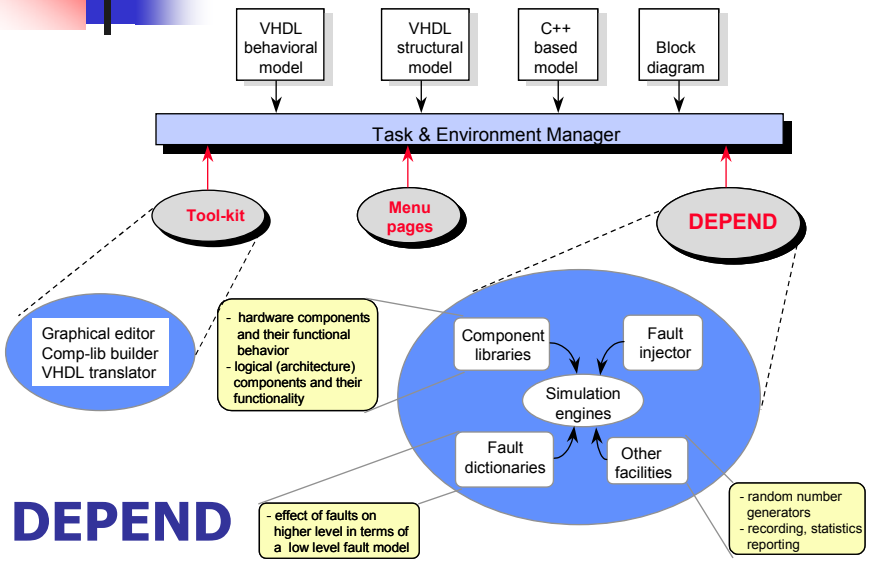
ITUA - Intrusion Tolerance by Unpredictable Adaptation



Algorithms and architectures for building intrusion-tolerant distributed systems

ILLINOIS

Modeling and Simulation



DEPEND

A simulation framework to support the design of systems for fault tolerance and high availability.

It takes as inputs both VHDL and C++ system description and produces as output dependability characteristics including fault coverage, availability, and performance.

(License to several companies and employed to simulate a number of industrial Systems, e.g. Integrity S2 from Tandem)

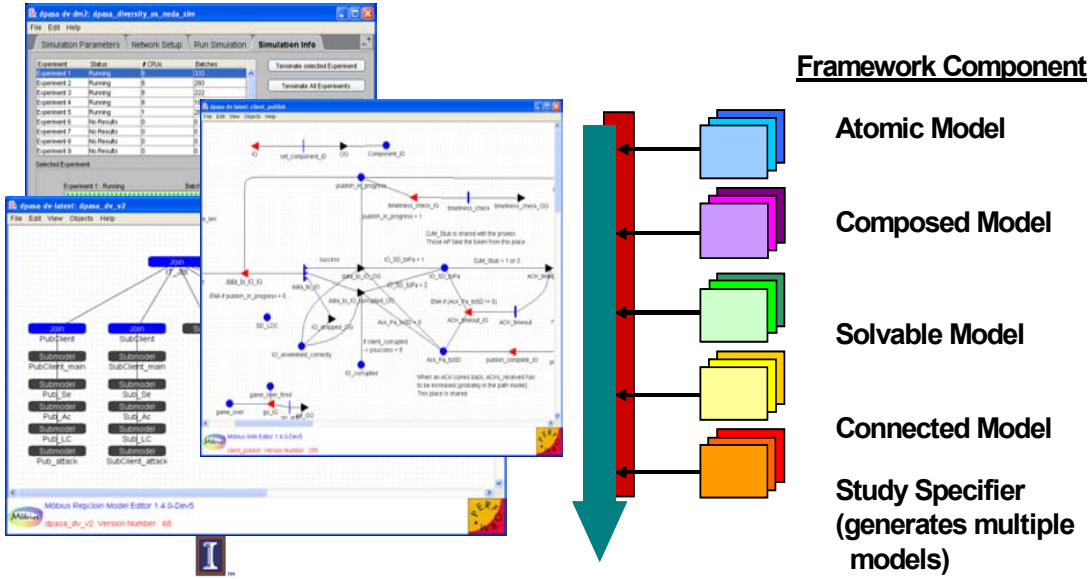
Möbius

An Extensible Modeling Tool for Quantifying Dependability, Security, and Performance

Features:

- Multiple model representation techniques facilitate modeling of hardware, software, protocols, and application in a unified manner
- Unified representation of dependability, security and performance attributes
- Integrated analytical/numerical and simulation-based solution

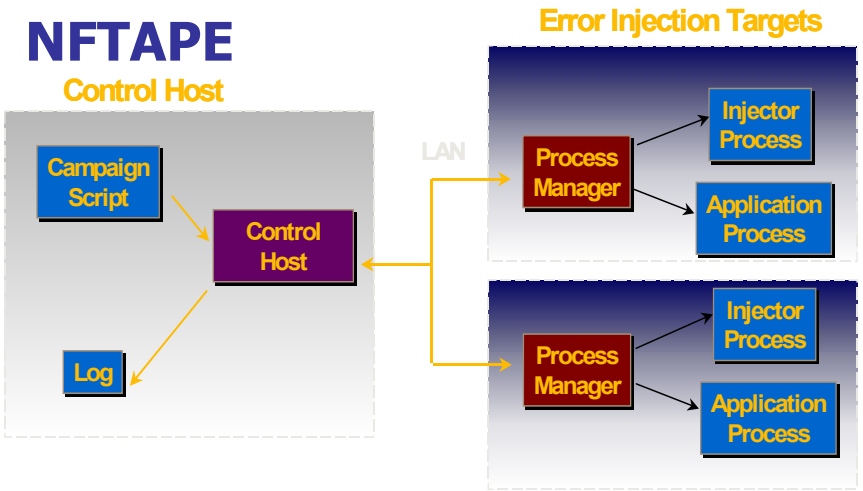
(Licensed by over 190 institutions)



ILLINOIS

Experimental System Evaluation and Benchmarking – Fault Injection

NFTAPE



A software framework for conducting automated fault/error injection-based dependability characterization; Evaluation/benchmarking of:

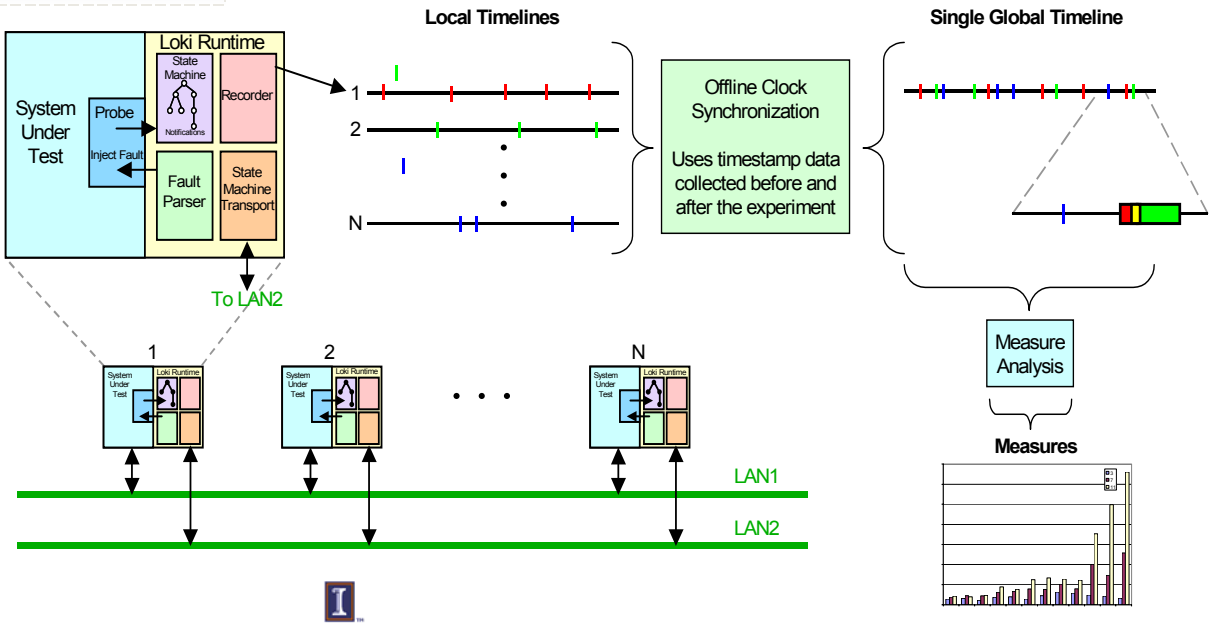
- fault-tolerant systems, e.g., Tandem FT-platforms;
- operating systems, e.g., Linux kernel on Pentium and PowerPC
- applications, e.g., call processing, space-borne software
- security violations due to errors, e.g., FTP, firewall facilities

(Licensed to multiple institutions, including JPL-NASA)

Loki

A software fault injector for distributed systems in which the introduction of faults is triggered based on the global state of the system.

Evaluation of large-scale distributed systems, e.g., a group membership protocol in Ensemble, and correlated network partitions in Coda distributed file system.



ILLINOIS

Operational System Data Analysis

Failure Data Analysis

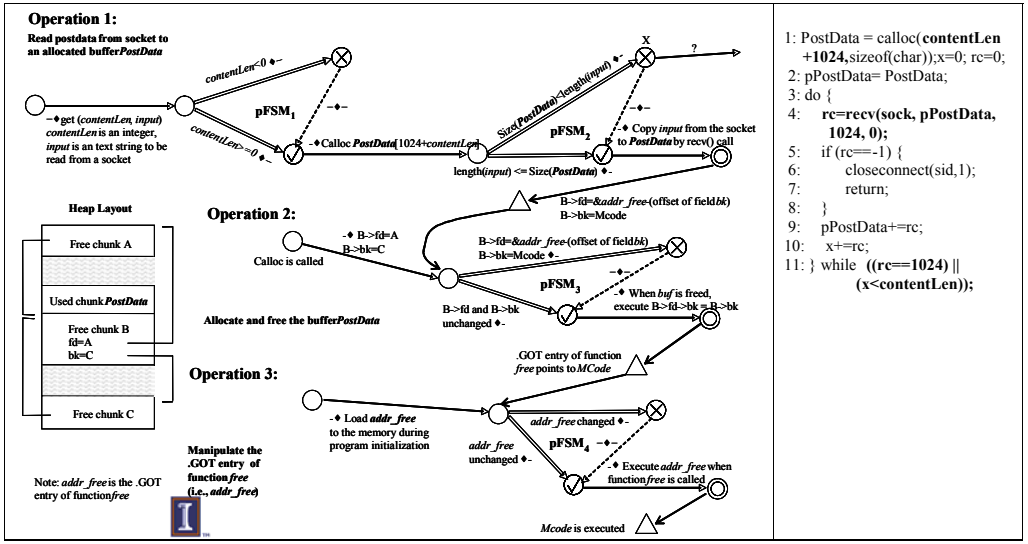
		LAN of Windows NT Machines		Internet
Data type and number of machines		System logs from 503 servers	System logs from 68 mail servers	Web site access success/failure logs from 97 most popular Web sites
Period for data collecting		4 months	6 months	40 weeks
Failure context		Machine reboots logged in a server system log.	Machine reboots logged in a server system log.	Inability to contact a host and fetch an HTML file from it.
Availability	**System perspective**	99%	99%	N/A
	User perspective (user gets expected service)	N/A	92%	99%
Downtime (median)		N/A	0.2h	1h (45%) 4.5h (49%) 53h (6%)
Comments		Indication of error propagation across the network		A few major network-related failures made nearly 70% of the hosts inaccessible

Security Vulnerability Analysis

Combine an analysis of data on vulnerabilities with a source-code examination to develop FSM model to depict and reason about security vulnerabilities.

- Exploits must pass through multiple elementary activities
- Multiple vulnerable operations on several objects are involved in exploiting a vulnerability,
- FSM model allows specifying logic predicates that need to be met to ensure security.

- Impact of workload on hardware and software fault tolerance
- Characterization of error latency
- Failure characterization of UNIX and Windows NT servers
- Reliability of Internet hosts



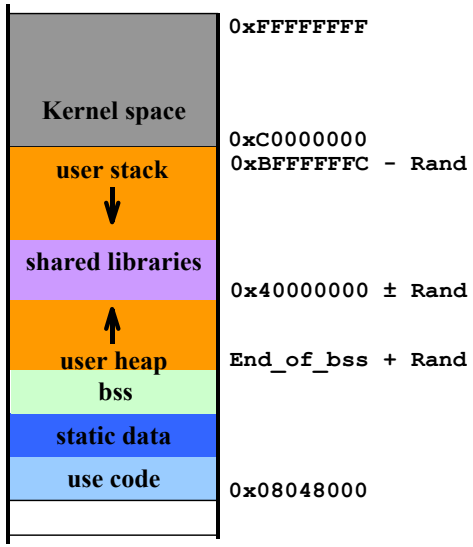
Security Vulnerability Avoidance and Runtime Protection

Vulnerability Avoidance

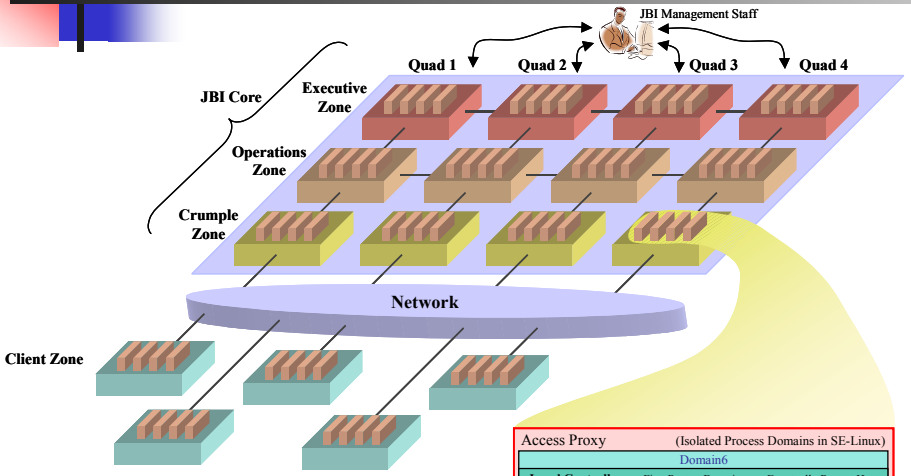
- CERT advisories indicate
  - ≥ 66% vulnerabilities due to pointer taintedness
  - ≥ 33% due to errors in library functions or incorrect invocations of library functions
- **Pointer Taintedness** – a unified basis for reasoning about security vulnerabilities
  - A pointer is tainted if a user value, including a return address, is derived directly or indirectly from the user input.
- **Pointer taintedness semantics** applied to formally reason and extract security specifications of library functions
  - Exposes (and removes) security vulnerabilities

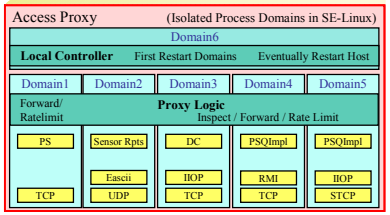
Transparent Runtime Randomization

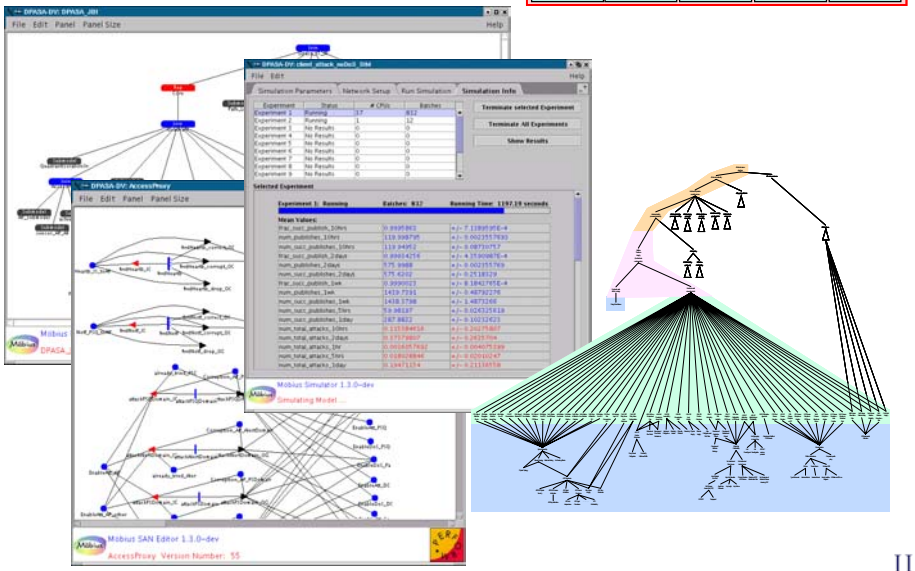
- Protects against about 60% of security attack reported by CERT
- Breaks the attacker's assumptions on the fixed memory layout of the target system
- Makes it impossible to correctly determine location of critical application associated memory regions essential in designing and launching a successful attack
- Converts an attack into application crash
- Program initialization overhead – 1% to 6%
- NO runtime overhead
- Memory cost – extra copy of GOT(global offset table) 200 Byte to 3.5 KB



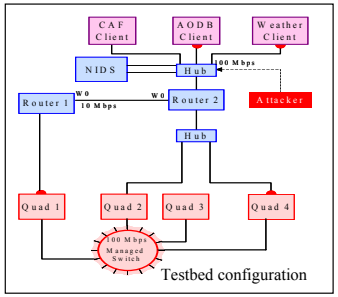
DPASA – Designing Protection & Adaptation Into a Survivability Architecture



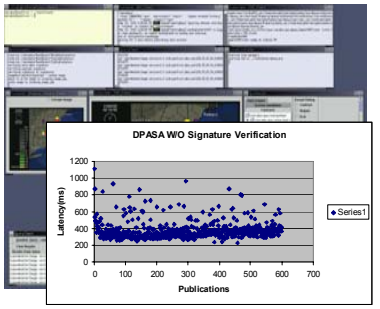




Enough proof of concept implementations developed to show 3 AFRL clients running a simplified scenario over 4 quad core



Testbed configuration

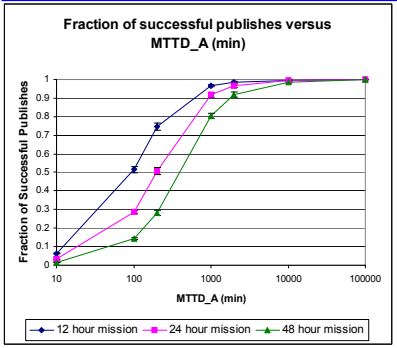


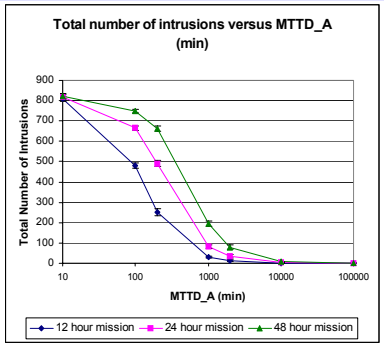
Screenshots and performance graph

Demonstrates

- ADF based protection
- DJM: signing and signature checking, authentication, RBAC, semantic and behavioral checks, FT protocols
- Adaptive response: rapid reaction, IO rejection, quad isolation, dynamic clients, fall back

■ The DPASA IT-JBI design provides critical functionality with high probability even when under heavy successful attack.





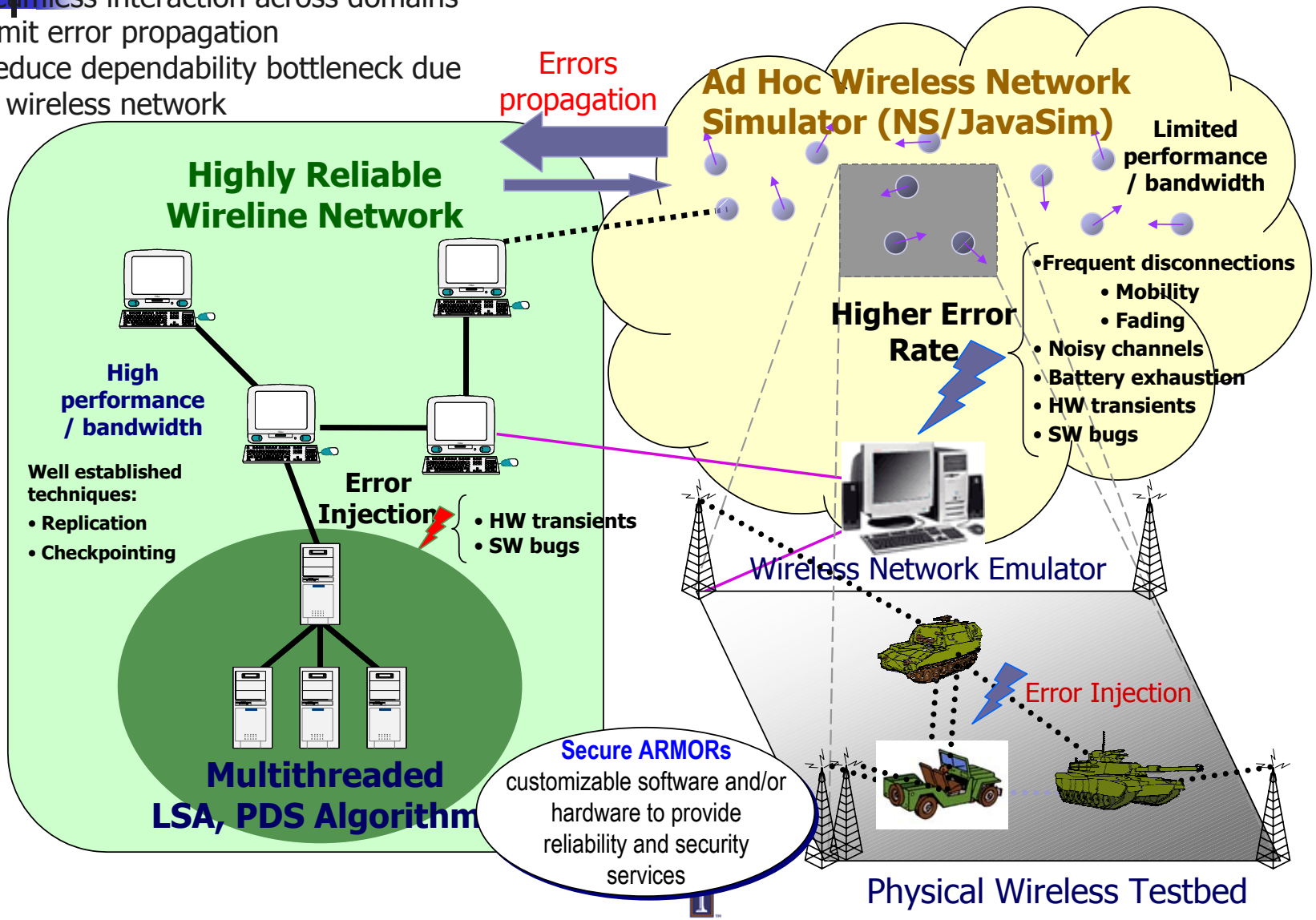
■ 98% of all publishes successful when new vulnerabilities are discovered, on the average, once a day or less often during a 12-hour mission. (An extremely high new vulnerability discovery rate; CERT data suggest $MTTD_A$ ~ 6000 min.)

■ At this new vulnerability discovery rate, system provides correct functionality even when about 10 intrusions occur during a 12-hour mission.

ILLINOIS

Secure ARMORs – Seamless Dependability & Security Testbed

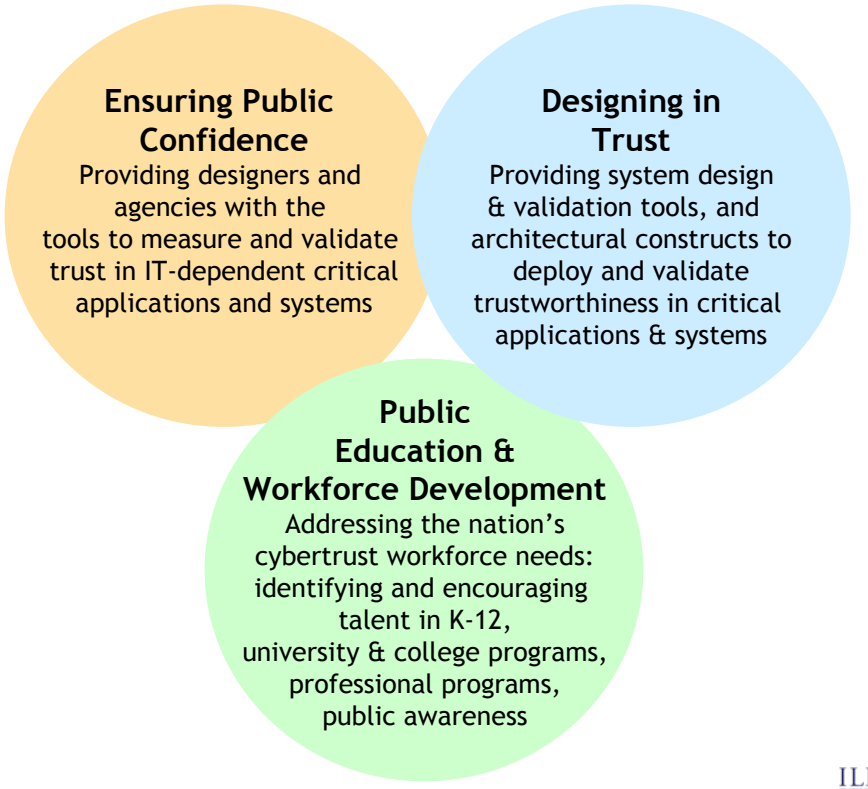
- Seamless interaction across domains
- Limit error propagation
- Reduce dependability bottleneck due to wireless network



The Information Trust Institute (ITI)

Trust is about **public confidence** –
It is about **security**, but also **correctness**, **reliability**, **availability**, and **survivability**

- The loss of public confidence in or access to critical systems can be economically devastating:
- Two weeks after 9/11: >$10B and 100,000 jobs lost in airline industry
- Downtime costs per hour: brokerage operations: $6.45 M, credit card authorization: $2.6 M



ITI: **Integrated private/public R&D and workforce development** to foster technology transfer, new industry, products, and services in order to provide design and validation tools and architectural constructs needed to ensure and justify **public trust** in **critical applications and systems**

ILLINOIS

Research in Dependable and Secure Systems at Illinois

Faculty

- Microprocessor architecture – S. Patel, N. Carter
- Hardware checkpointing – J. Torrellas
- Formal methods – J. Meseguer
- Test and VLSI design – J. Patel
- Storage systems – Y. Zhou
- Mobile computing – N. Vaidya
- Simulation – D. Nicol
- Modeling and design – W. Sanders
- Optical networks – S. Lumetta
- Measurement and design – R. Iyer
- Benchmarking and design – Z. Kalbarczyk

Major Partners

- IBM – benchmarking
- SUN – next generation supercomputer
- BBN – Intrusion-Tolerant Computing and Adaptive Systems
- Motorola – wireless communication and computing
- HP – utility computing
- Boeing – trusted software
- JPL – next generation space borne computing