EECE 499/693: Computers and Safety Critical Systems

4 Design of Fail-Safe Computer System

B. Dual Redundant Architecture

Instructor: Dr. Charles Kim

Electrical and Computer Engineering
Howard University

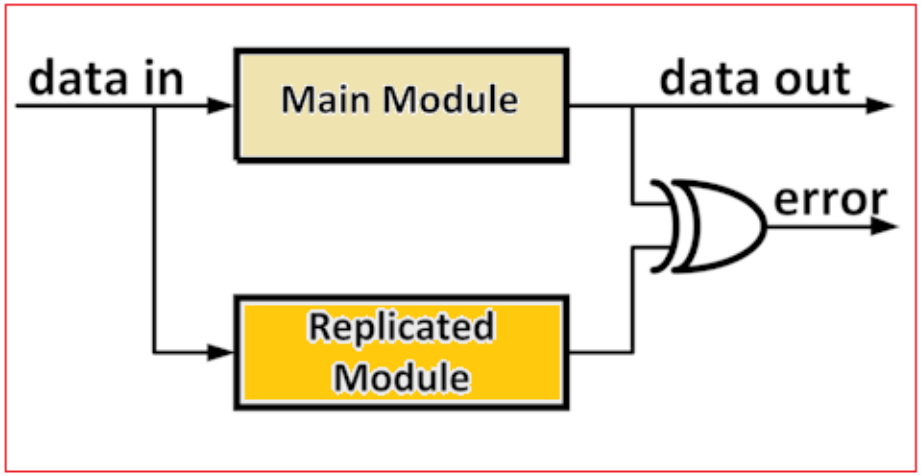
www.mwftr.com/CS2.html

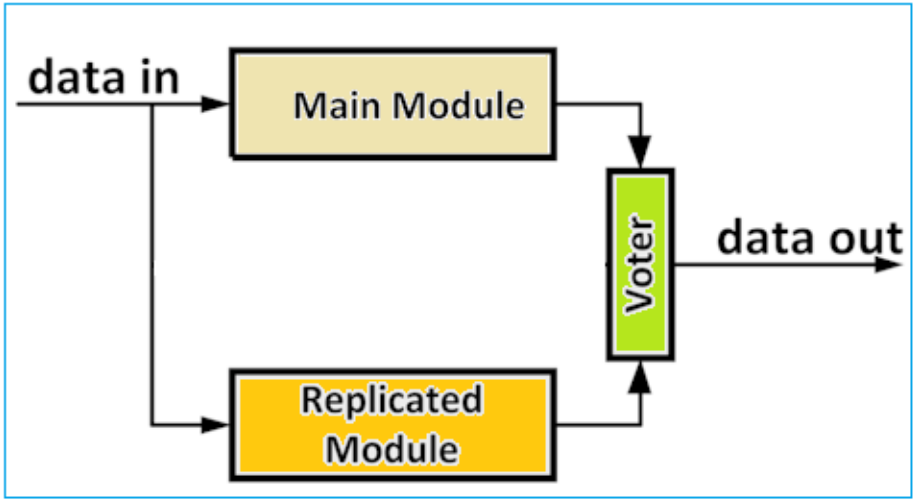
Simplex Architecture - Review

- A simplex system: “a system which does not employ redundancy” whether it be a basic system or a fail-safe system
- We have discussed how this simplex system can behave fail-safe under fault and failure events in each of the component of the example system
- Sensor/Actuator failure detection by Simplex Architecture
  - Single sensor failure detection by estimated correct sensor value
  - **Problem:** Estimations are too coarse of take too long to compute
  - **Problem**: Estimations may not be known in advance
- Computer Hardware Fault/Failure detection by Simplex Architecture
  - Watchdog Timer
  - S/W based test for CPU, Memory, and I/O devices
  - **Problem**: S/W based tests may take too long over a large number of frames so that failures may surface before the faults may be diagnosed and found
  - **Problem**: Some Off-the-shelf system may not allow to incorporate s/w-based diagnostics

Dual Redundancy

- When failure of a simplex component cannot be reliably detected, it is necessary to adopt pure brute force - Dual Redundancy
- Dual Redundancy: Two identical components are employed and run in parallel
- Operation (Under the assumption of single failure)
  - Matched outputs: no failure
  - Unmatched outputs: a failure





Dual Redundancy – Sensors

- Background:
  - Uncertainties and complexities of detecting simplex sensor failures
- Simple approach:
  - Duplication of the sensors
    - **One**: monitoring and control
    - **Second**: A reference that provides the known value for use in failure detection
  - Both are read by a single computer
  - They are compared

Dual Redundancy – Sensors

NASA Technical Reports Server (NTRS)

Providing Access to NASA's Technology, Research, and Science

search a

BASIC SEARCH | ADVANCED SEARCH | ABOUT NTRS | NTRS NEWS | OAI HARVEST | TUTORIALS | SEARCH TIPS

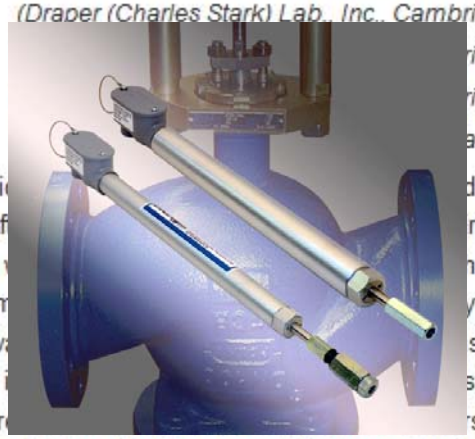
Record Details

‹ Return to Search Results ‹ Previous Record | › Next Record › Printable View

Text Size + − Record 1 of 1 Send Share

Dual redundant sensor FDI techniques applied to the NASA F8C DFBW aircraft. [Failure Detection and Identification

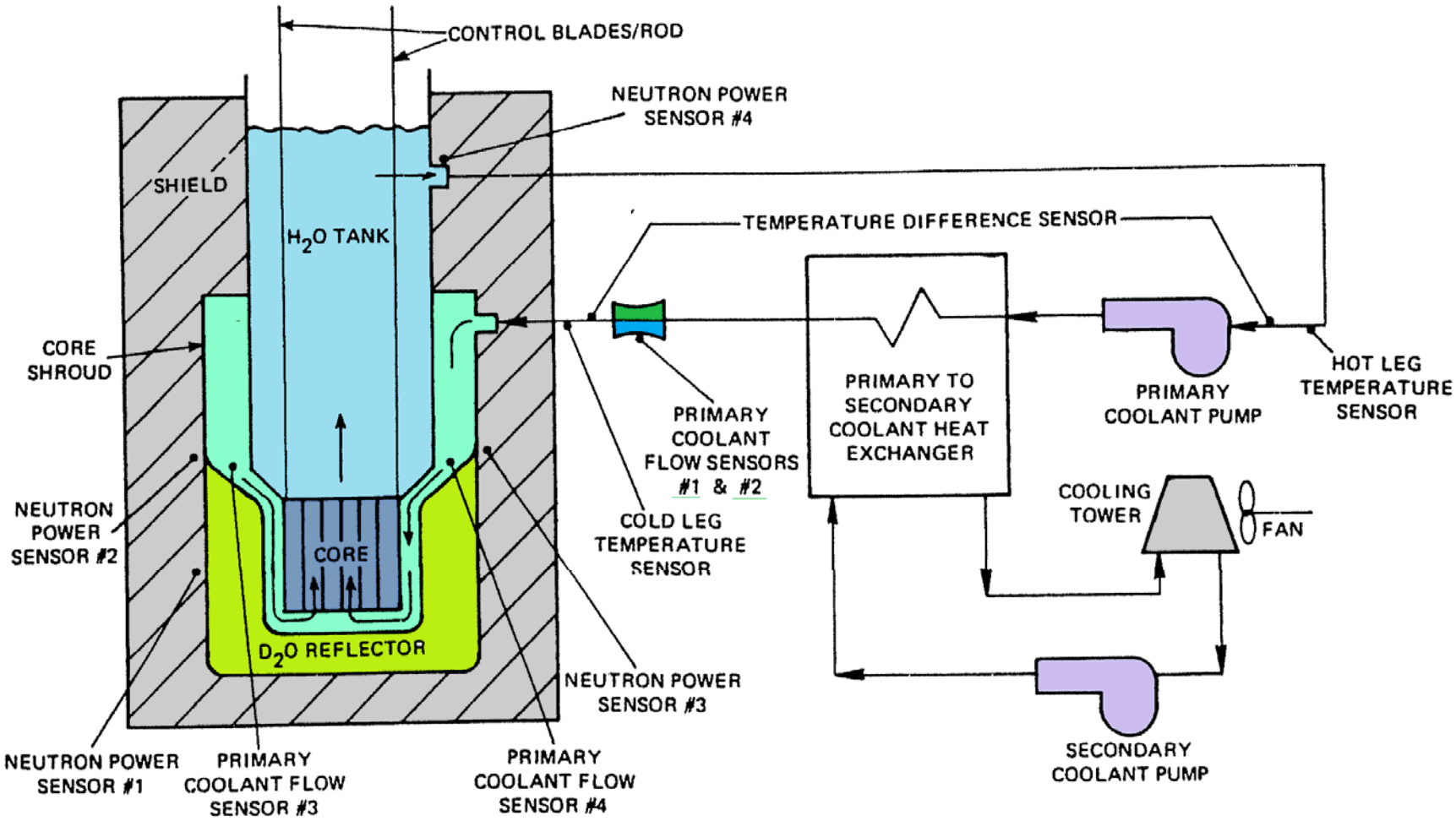
Author and Affiliation: Desai, M. N. *(Draper (Charles Stark) Lab., Inc., Cambridge, MA, United States)*;
Deckert, J. C. *(Draper (Charles Stark) Lab., Inc., Cambridge, MA, United States)*;
Deyst, J. J. *ridge, MA,*
Willsky, A. S. *ridge, MA,*
Chow, E. Y. *ambridge,*



Abstract: An onboard failure detecti… dual redun…
fly-by-wire (DFBW) aircraf… r of senso…
direct redundancy trigger … e outputs…
the failed sensor is accom… y that exis…
relationships among the v… struments…
failures, common to both i… shed by u…
analytic redundancy failur… s. The ba…
of the measurement of a variable using the suspect instrument with another obtained using other instrument types.

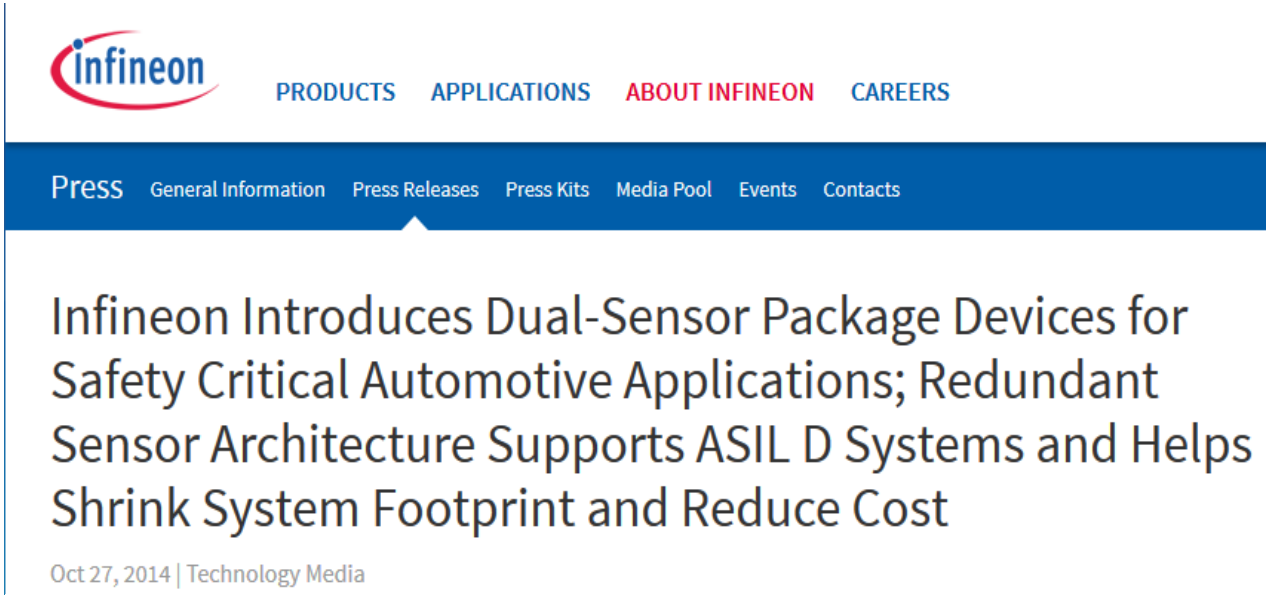
Publication Date: Jan 01, 1976

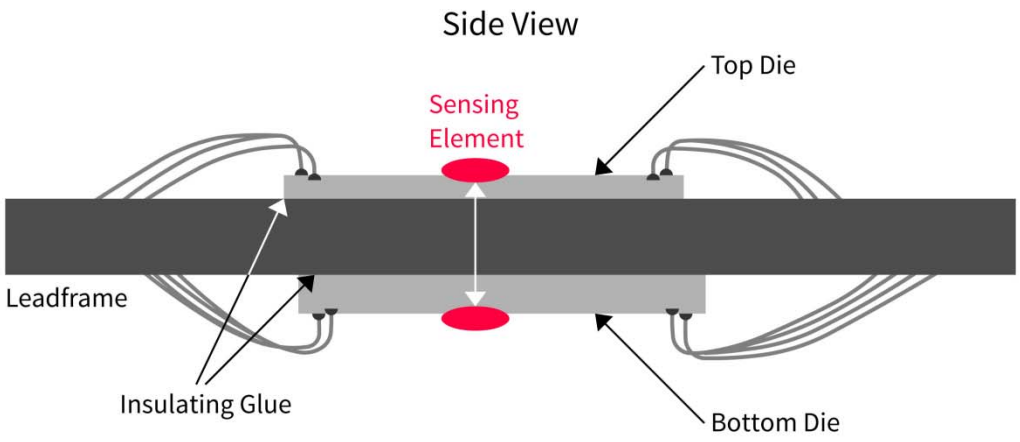
Dual Redundancy – Sensors



Simplified schematic diagram for the nuclear reactor.

Dual Redundancy – Sensors

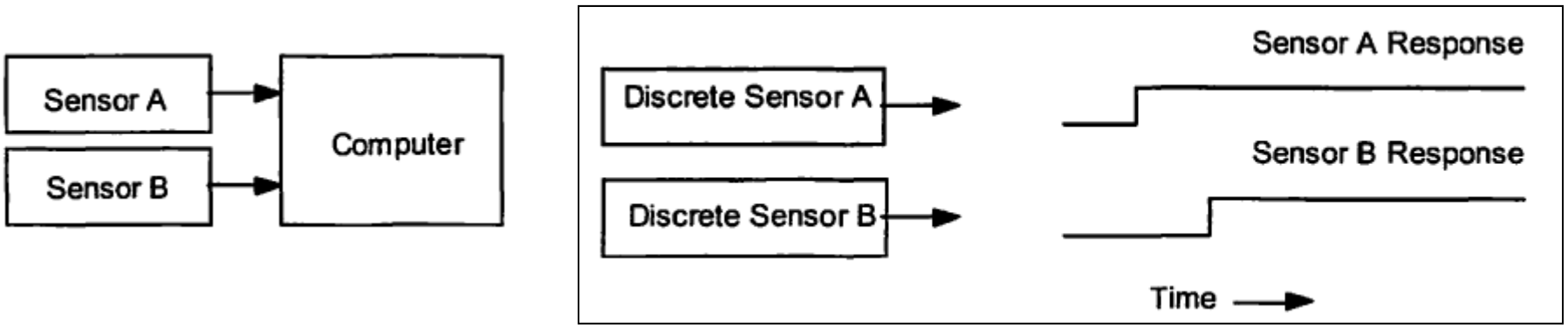




- **ASIL** (Automotive Safety Integrity Level): Risk classification scheme defined by ISO 26262 – Functional Safety for Road Vehicles standard.
- 4 grades: ASIL-A (lowest), -B, -C, and –D (highest)
- **ASIL D**: Highest class of initial hazard & most stringent safety measures applied to avoid unreasonable residual risk.

Dual Redundancy – Sensors: Time Skew

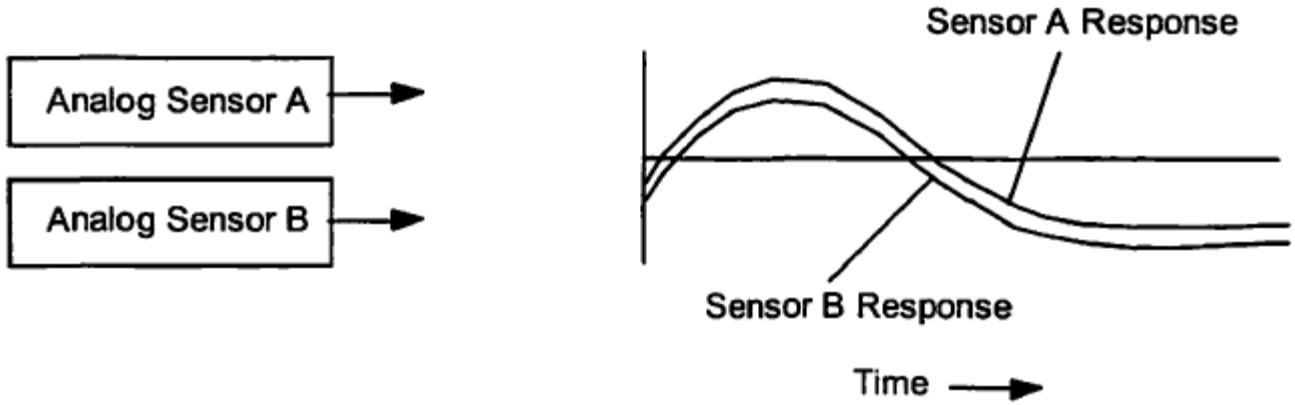
- Problem of **Skew in Time** in the Dual Sensor Architecture



- **Causes of the Skew** in Time in the sensor outputs
  - Two sensors are placed in the physically different places
  - Two sensors are constructed differently
  - Two sensors are calibrated at different times
- **Solution for the Skew** in Time
  - Handling by software in consideration of the timing of comparison

Dual Redundancy – Sensors

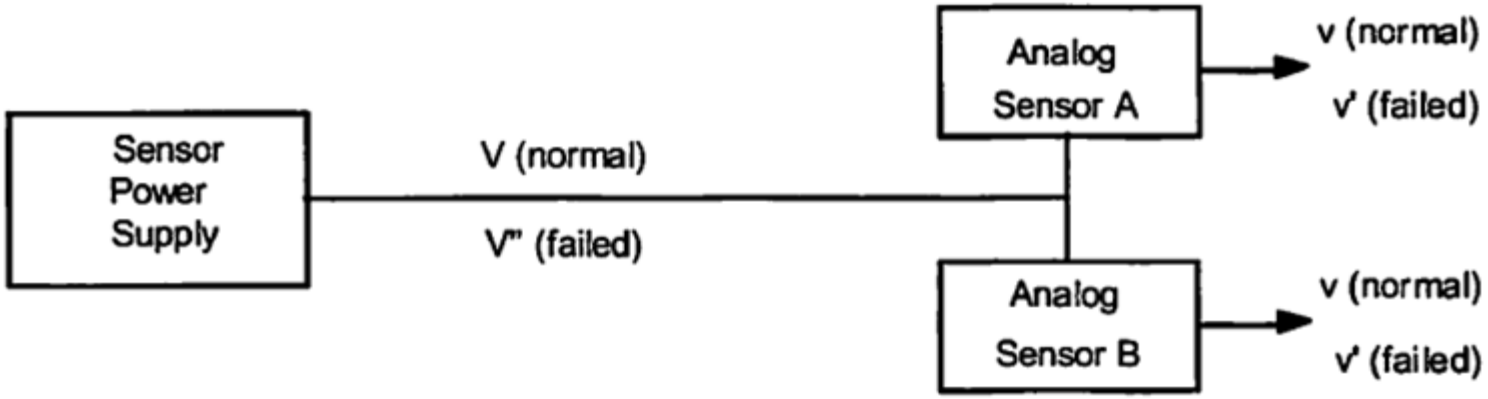
- Problem of Analog Value **Difference in two sensors**



- **Causes** of the different sensor outputs
  - Two sensors are constructed differently
  - Two sensors are calibrated at different times
- **Solution** for the Skew in Time
  - Handling by software in consideration of the threshold in determining the values

Single Points of Failure in Dual Sensors

- Dual Redundant Sensors & Simplex Computer (& Simplex Power supply)
- Simplex component failure may brings in **matching but incorrect results** in the dual sensors
- Example: Failure in the Simplex Interconnect

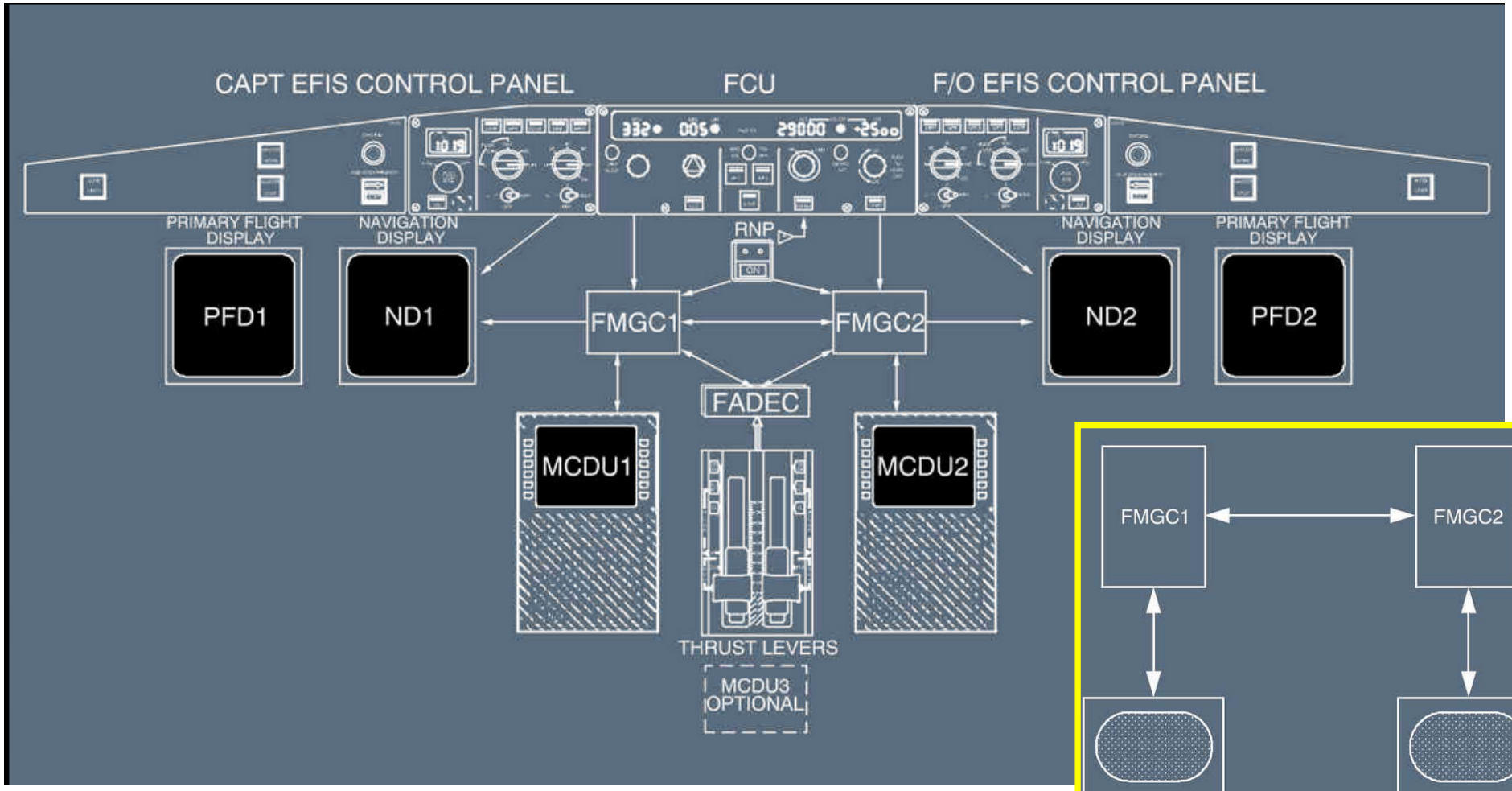


- **Point:** Failures in the simplex elements may compromise any safety margin gained in using dual sensors

Dual Redundancy – Computer Hardware

- Adoption of dual computer redundancy
  - Hardware single points of failure is unacceptable
  - Failure detection speed is important
- Operation
  - Each computer functions identically
    - No failure: two produce matching outputs
    - Failure: Outputs do not match
- Dual Architecture
  - Number of ways

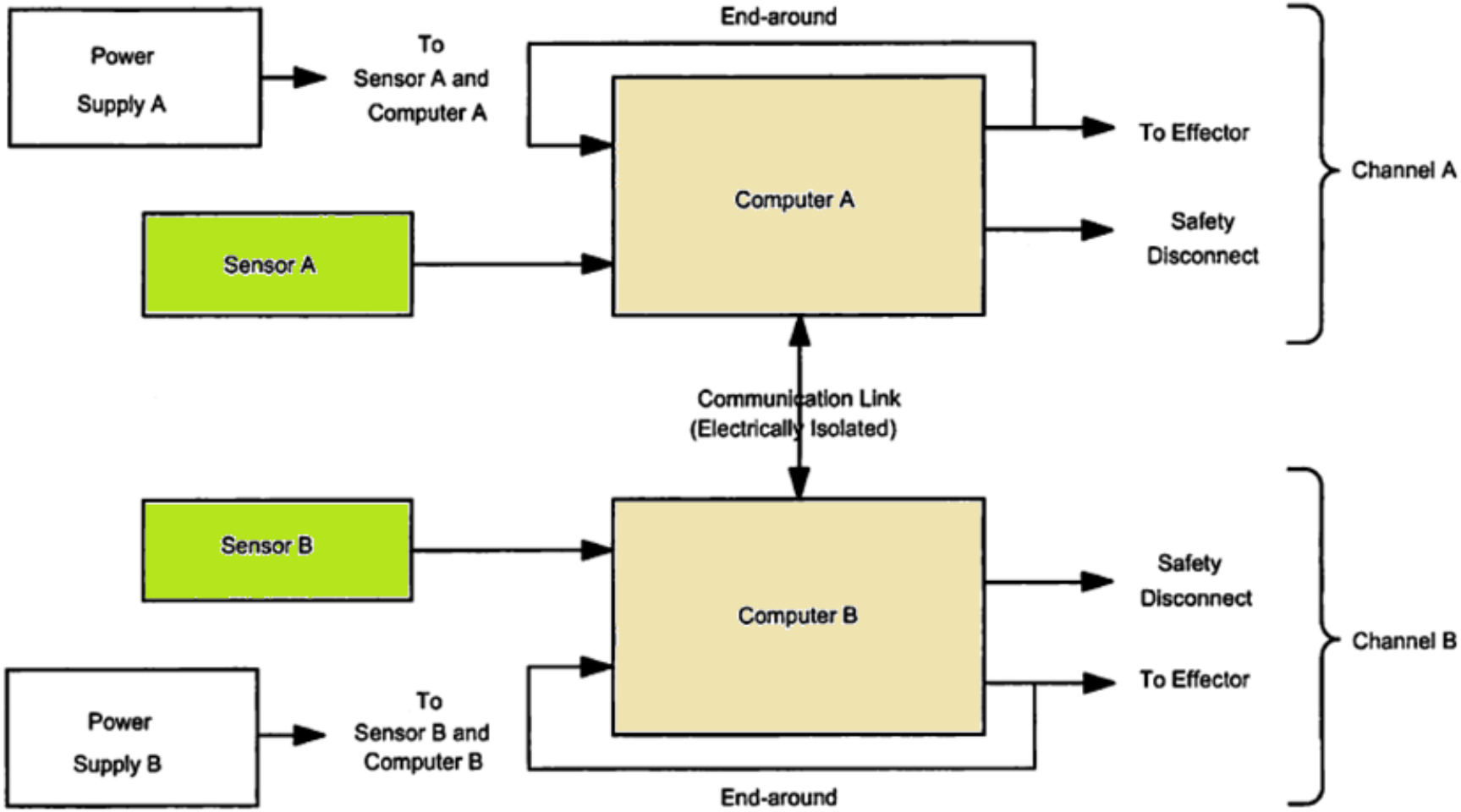
Dual Redundancy – Computer Hardware



Source: http://airborne.blogspot.com/2013/05/auto-flight-general.html

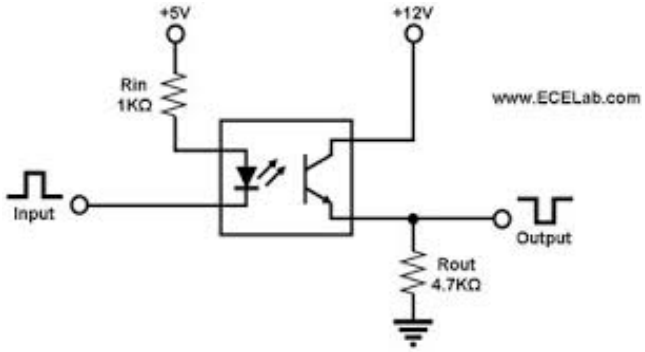
Dual Computer Architecture

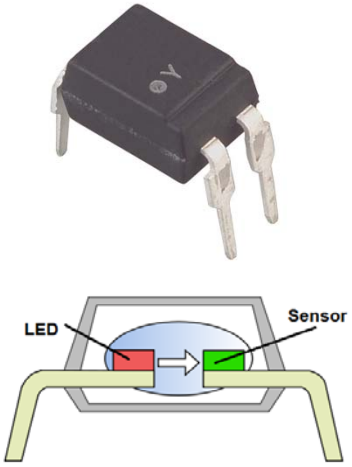
- Computer hardware, power supply, and interconnects (and sensors) are all duplicated
- Each of the groups is referred to as a channel



Dual Architecture

- Assumption:
  - 1. Hardware in the channels is **independent** → A hardware failure in a channel has np effect on the correct performance of other channels
  - 2. The communication path is **electrically isolated** from the computers → a hardware failure (such as a short circuit) in the connecting path will not propagate to computers
  - "Electrically Isolated": meaning?
    - Line connecting the two computers is transformer or opto-electronically isolated

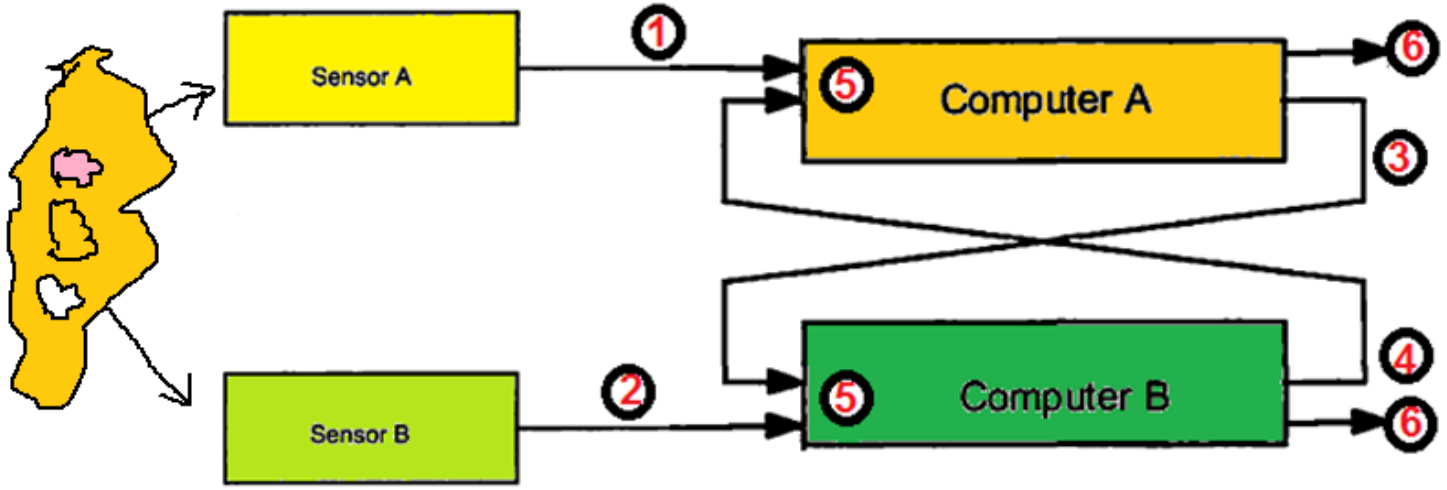




Dual Architecture --- Software Composition

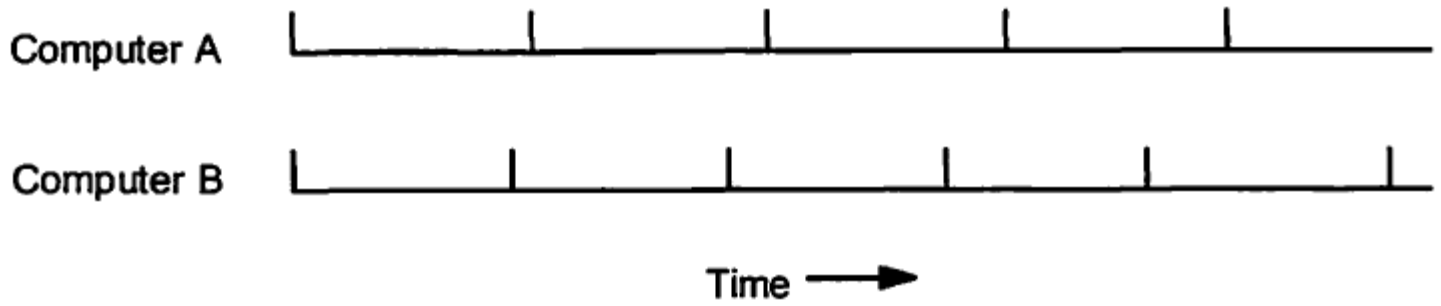
- Software Functions
  - **First function**: **Normal control and monitoring** with sensor data reading, effector value computation, and effector value out-putting
  - **Second function**: **hardware failure detection** for sensors and computer → **Our focus**
- Software composition for hardware failure detection

1. Computer_A reads sensor_A
2. Computer_B reads sensor_B
3. Computer-A sends its sensor value to Computer_B
4. Computer_B sends it sensor value to Computer_A
5. Both Computers compare the two values
6. Declare Normal or Failure



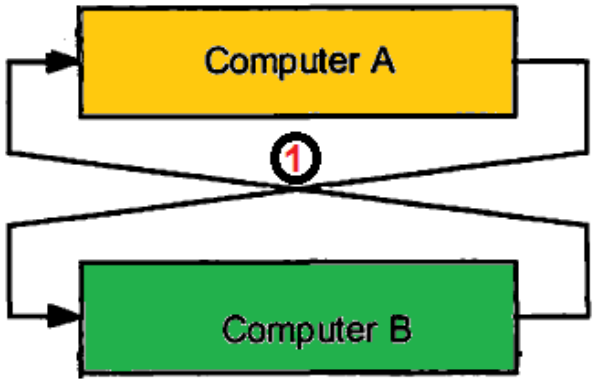
Dual Architecture ---- Frame Synchronization

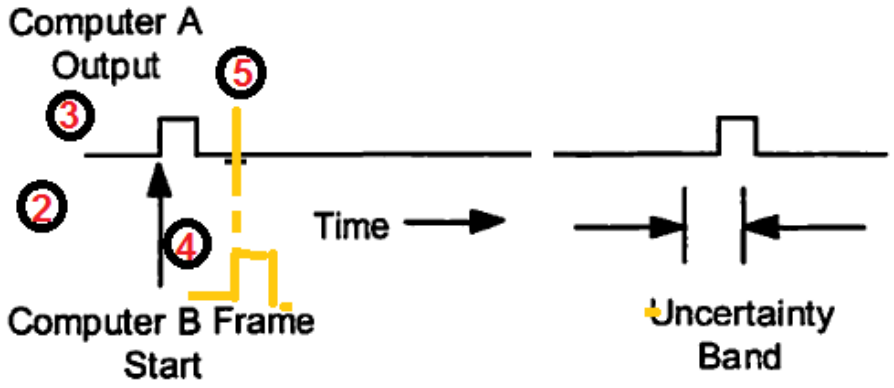
- Frames of two computers are each controlled by **internal clock**
- Two internal clock **may have different rate** → Frames of two computers drift with respect to one another



- Fixing the frame drift problem
  - **Computer B's clock** as an independent check on Computer A's frame period

Dual Architecture --- Frame Synchronization





Maximum Error Allowed in the 2 Frames

1. Wire dedicated discretes from A to B and B to A
2. Computer B starts to continuously monitor the discrete channel
3. Computer A software generates a pulse and sends to the discrete channel
4. When Computer B sees the positive transition of the signal, it begins its frame including generation of a frame pulse similar to that of Computer A. As the end of the frame, Computer B will have completed its computations and returns to sampling of Computer A' s discrete signal so as to detect and synchronize on the next positive transition.
5. Computer A samples Computer B's frame pulse to verify that it matches its own.

Dual Architecture --- Software

- Simplex Software
  - Use of **identical software in dual hardware channels** sets up the possibility that a **single software fault can effect both channels**, eroding the safety benefit gained by the dual redundant hardware
- Measures and cures
  - Dual independent watchdog timer
  - Software failure detection
  - **Dissimilar software**
  - **Diversity** –functional diversity and design diversity
- Defense-in-Depth and Diversity (D3)

Defense in Depth

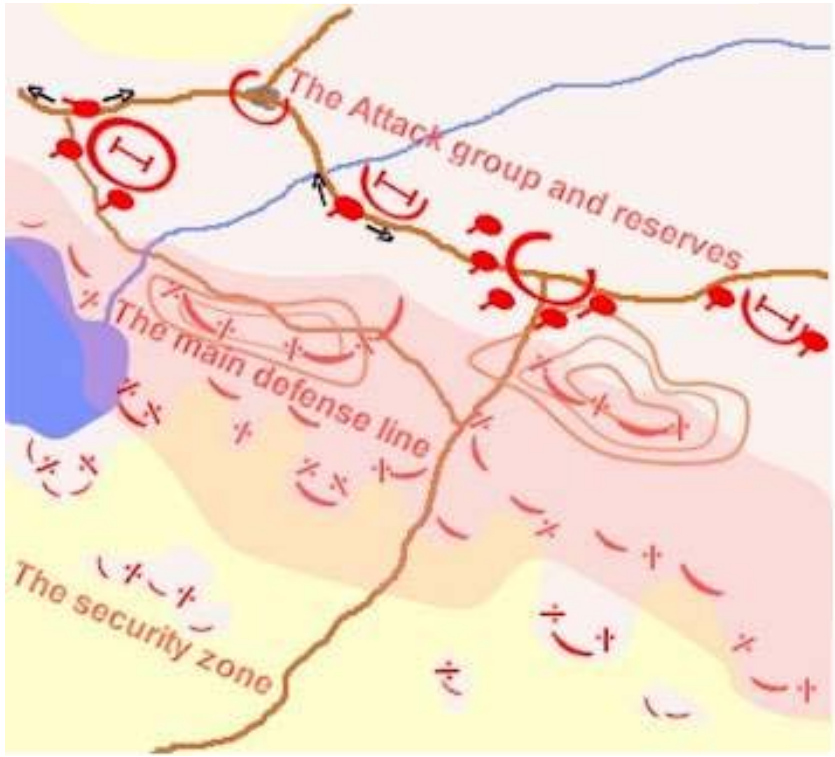
- Military Strategy
  - Front Line
  - Forward Defense
  - Defense-in-depth
- Industrial Use
  - Computing
  - Security
  - Nuclear Power
  - Aircraft
  - etc

Defense-in-Depth as Military Strategy

- Forward Defense --- Roman army
  - Garrison posts in Barbarian territory
  - Battle Fields – out of Roman territory
  - Expensive

- Front Line
  - Everything at the border line
  - Win or Lose
  - Maginot Line

- Defense-in-Depth
  - Thin Presence in the border line
  - Delay the advance of enemy
  - Strong defense line behind
  - Modestly expensive



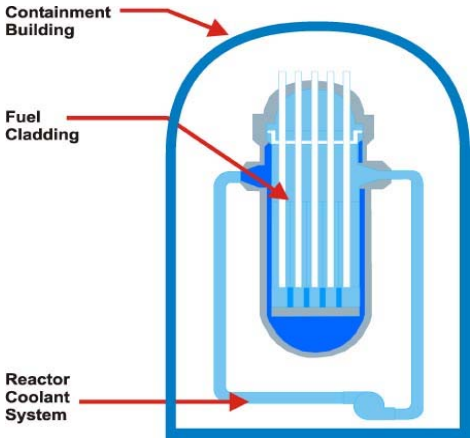
Defense-in-Depth in Information Assurance

- **Information assurance (IA) concept**
  - conceived by the National Security Agency (NSA) as a comprehensive approach to **information and electronic security**
  - **multiple layers** of security defense are placed throughout an information technology (IT) system
  - **provides redundancy** in the event a security defense fails or a vulnerability is exploited
- **Examples**
  - Physical security (e.g. deadbolt locks)
  - Authentication and password security
  - Hashing passwords
  - Anti virus software
  - Firewalls (hardware or software)
  - IDS (intrusion detection systems)
  - VPN (virtual private networks)
  - Logging and auditing
  - Biometrics
  - Timed access control
  - Exclusive Software/hardware

Defense-in-Depth in Safety-Critical Industry

- Aircraft:
  - emphasizes redundancy - a system that keeps working when a component fails - over attempts to design components that will not fail in the first place.
  - an aircraft with four engines will be less likely to suffer total engine failure than a single-engine aircraft no matter how much effort goes into making the single engine reliable.

- Nuclear engineering and nuclear safety:
  - practice of having multiple, redundant, and independent layers of safety systems for the single, critical point of failure – reactor safety system.
  - **Reactor Safety System**: reduce the risk that a single failure of a critical system could cause a core meltdown or a catastrophic failure of reactor containment.



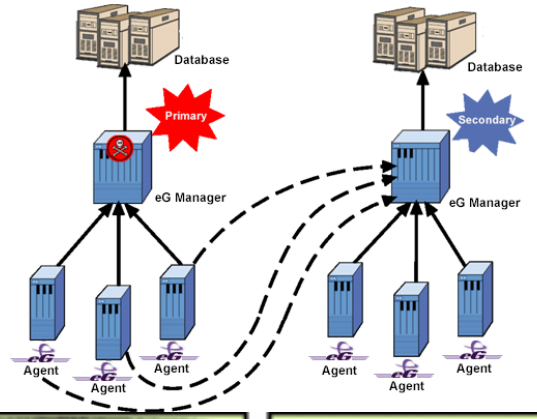
Defense-in-Depth and Redundancy

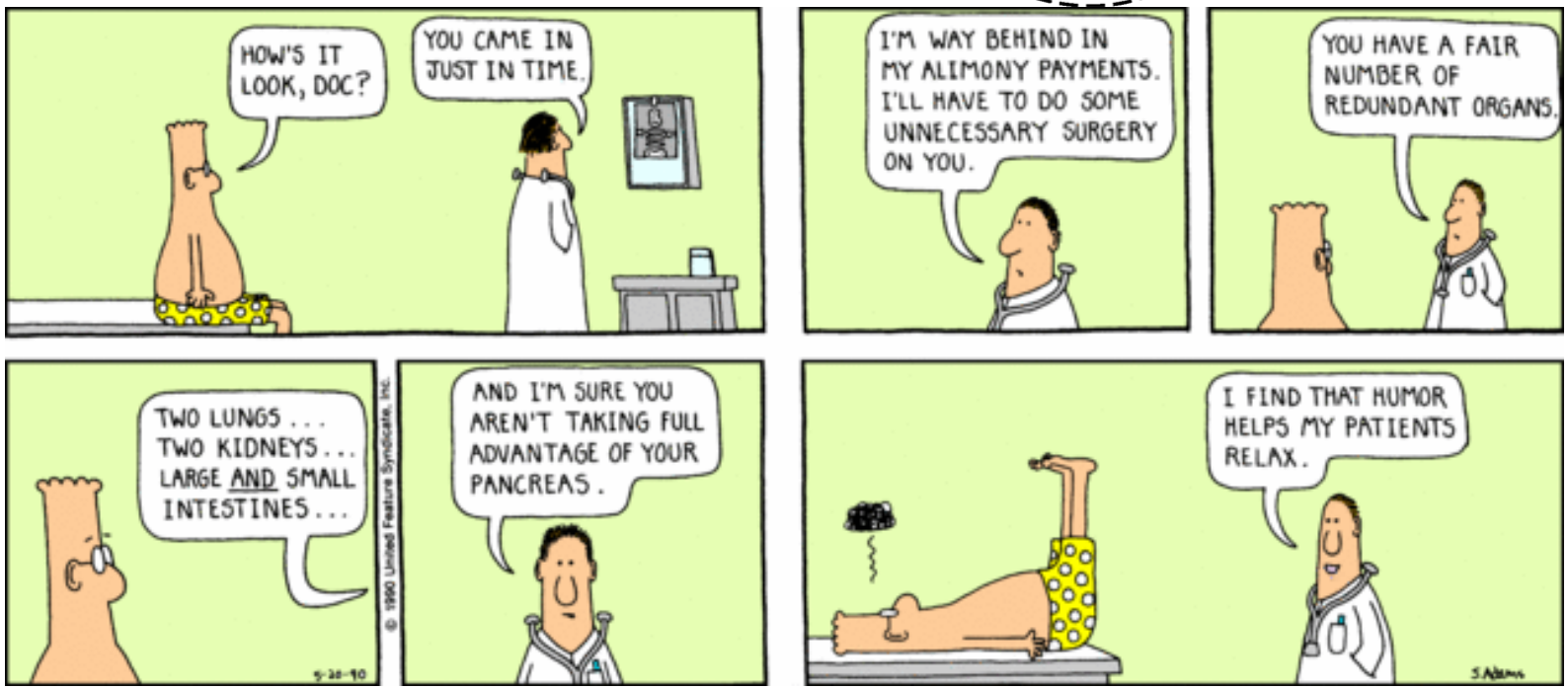
- Safety System must reliably satisfy the functional requirements
- Single-failure proof (no single failure is to prevent safety system actuation if needed, nor shall a single failure cause a spurious activation)
- How to achieve this goal?
  - By **Redundancy**
  - Achieve the functional goals in the presence of component failures
  - **Active** redundancy and **Standby** redundancy

Redundancy

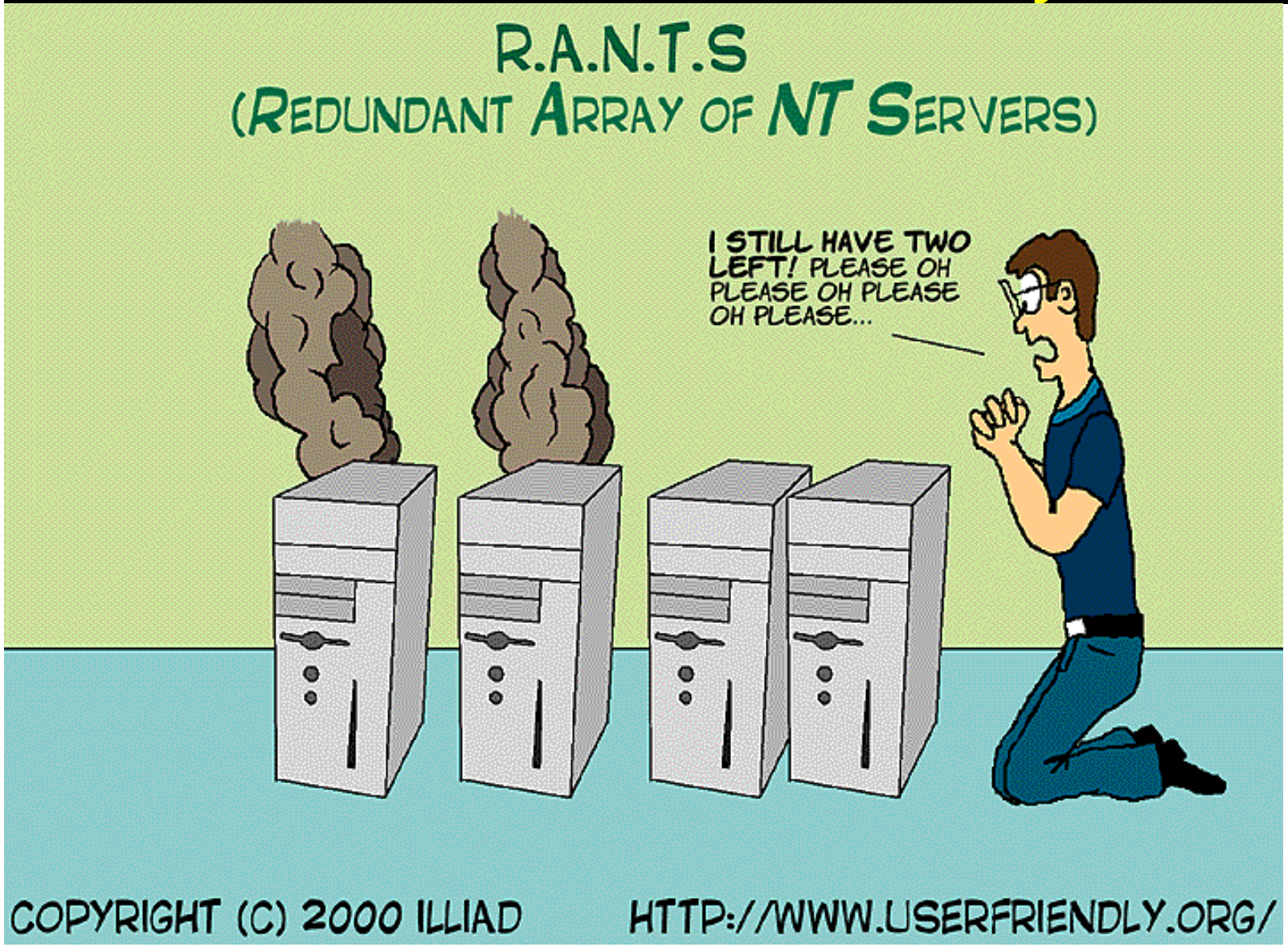
- Active Redundancy
  - Multiple identical components **operating in parallel**
  - The multiple outputs are compared or selected in some way to determine which outputs will be used
  - (ex) Boolean Logic; 2-out-of-3
- Standby (or backup) Redundancy
  - Make spares available **to replace** failed components
    - (ex)Backup generator
- Component duplication – Same function and identical component
  - Protection against independent failures caused by physical degradation (wear-out)

Redundancy in real life (Active? Standby?)

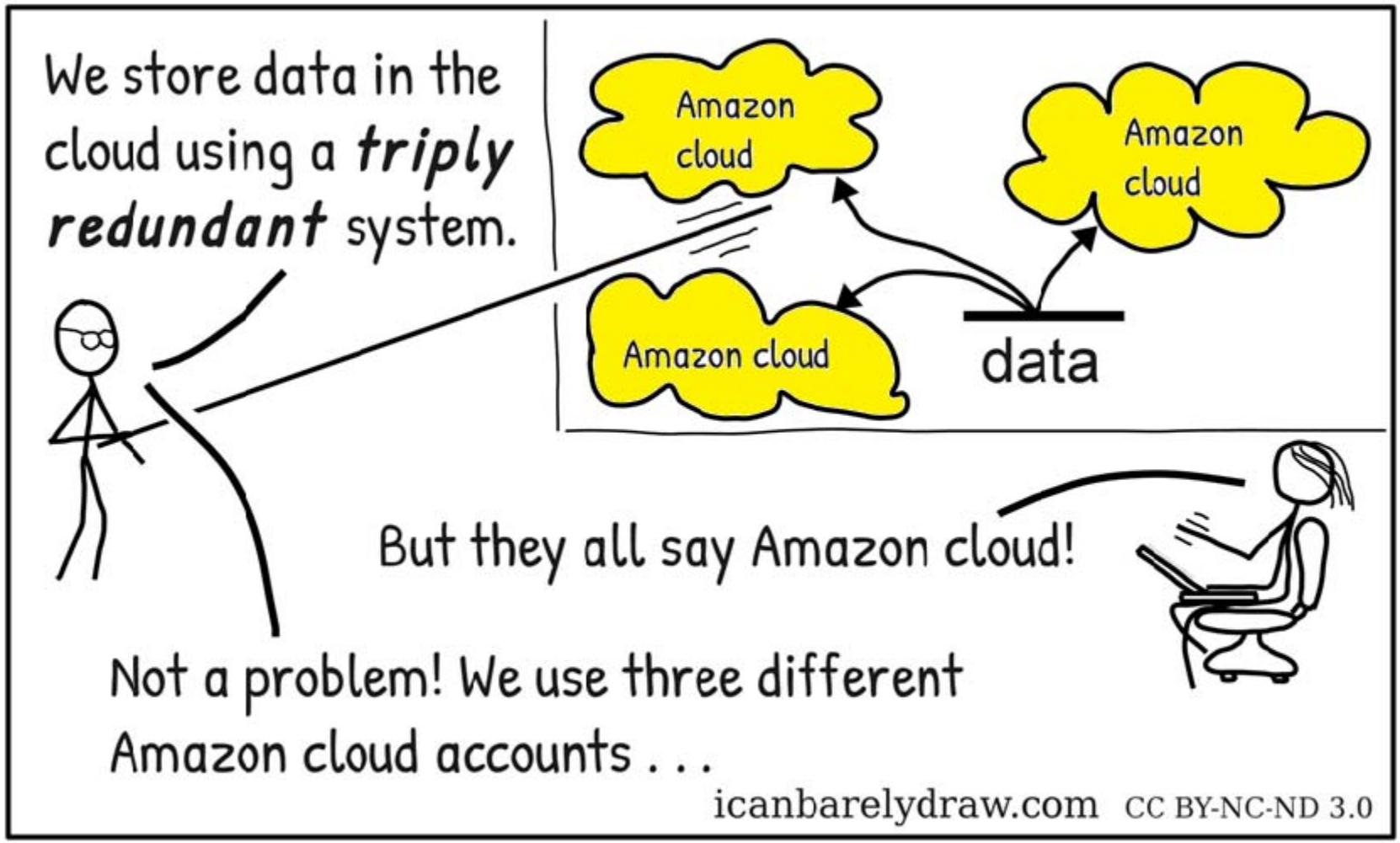




Problem in Redundancy



Vulnerability of Redundancy



Redundancy in the Cloud.

Common Cause Failure – Weakness of Redundancy

- The benefit of component duplication can be defeated by common-cause failure (CCF) or common-mode failures (CMF)
  - CCF: multiple components fail by the same cause
  - CMF: multiple components fail the same way
- CCF and CMF occur
  - because the assumption of independence of the failures of the components is invalid
  - Common external or internal influences
  - Design error

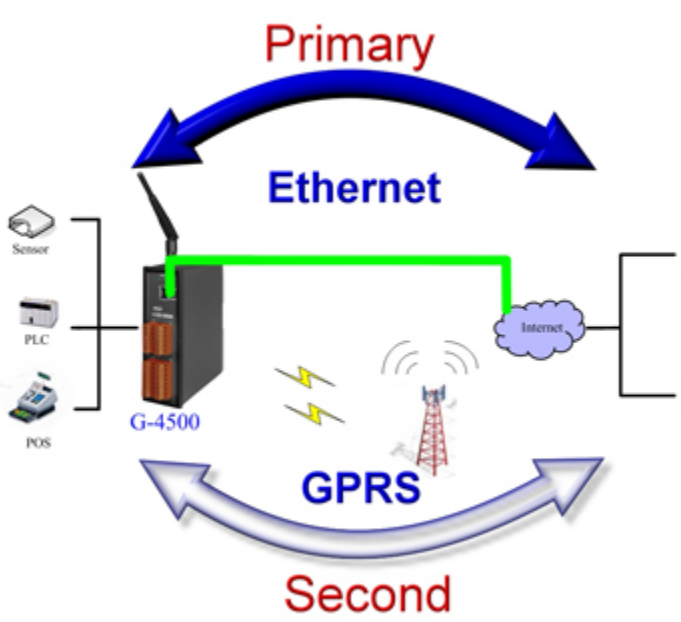
Protection against CMF - **Diversity**

- Design Diversity:
  - components with different internal design (but performing the same function) are used.
  - (ex) Multiple versions of software written from the equivalent requirements specifications – **same function by different algorithms** →(ex) two different ways of determining of two number are the same
  - (ex) Multiple different components differently achieving the design requirement

DIVERSITY

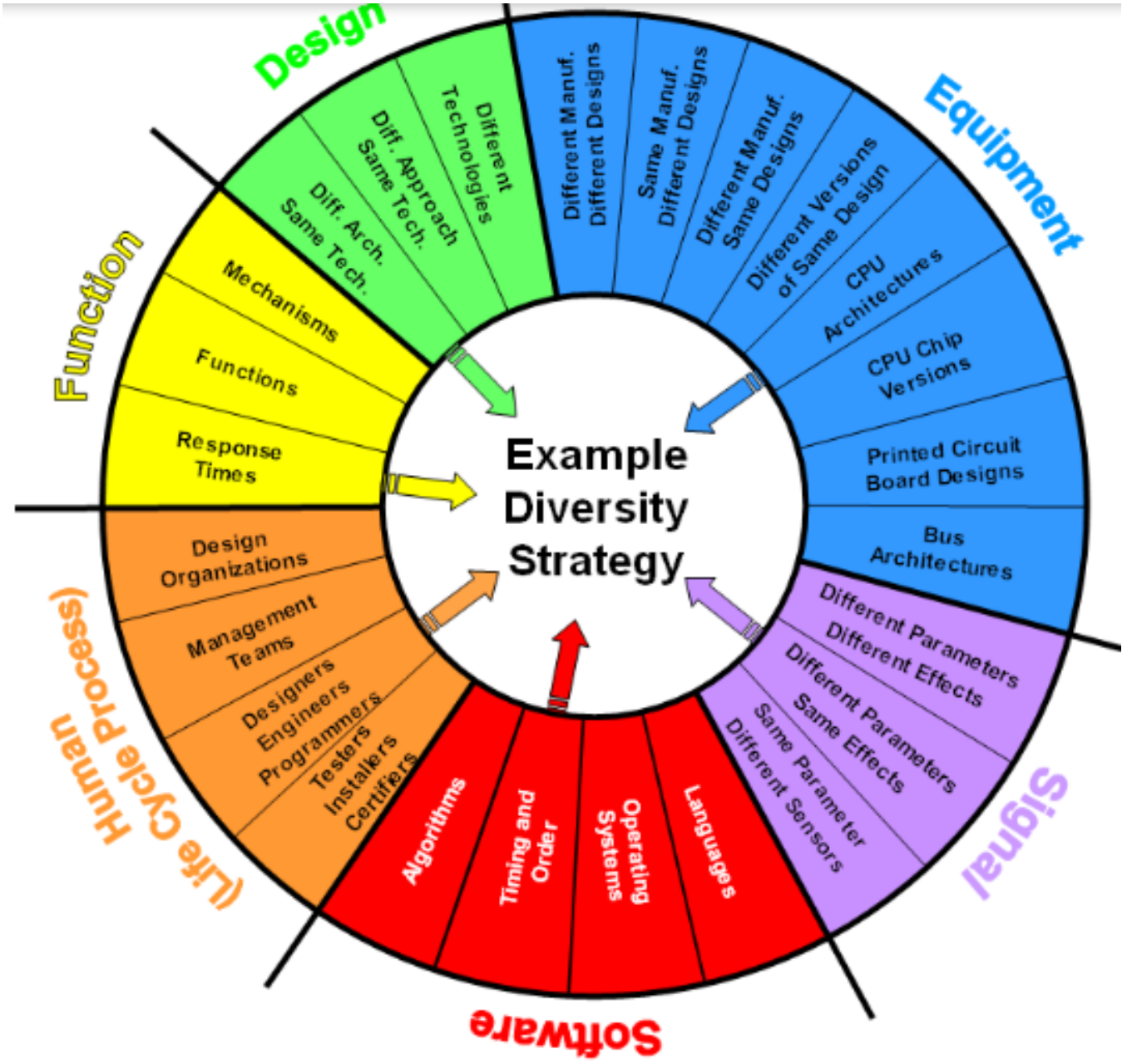
- Functional Diversity
  - Components made by **different requirements** perform **different functions at the component level** while satisfying **the upper level system requirements**
  - Different Principle of operation or physical principles to satisfy the same or different system-level requirements
  - (ex) one program checks if two numbers are equal; another program selects the larger of 2 numbers
  - (ex) One uses control rods to trip a reactor (based on the ratio of reactor power and flow); another uses Boron concentration to trip a reactor (based on coolant temperature)
- Most important issue: **Independence**

Diversity Everywhere



- GPRS: General Packet Radio Service – mobile data service on 2G and 3G Cellular Communication System

NRC D3 Strategy Examples



D3 Guidelines in Nuclear Industry

- NUREG/CR-6303, “Method for Performing Diversity and Defense-in-Depth Analyses of Reactor Protection Systems,” December 1994.
- NUREG-0800, Standard Review Plan, BTP 7-19, “Guidance for Evaluation of Defense-in-Depth and Diversity in Digital Computer-Based Instrumentation and Control Systems,” March 2007.
- U.S. Code of Federal Regulations, Title 10, Energy, Part 50, Section 62, “Requirements for Reduction of Risk from Anticipated Transient Without Scram (ATWS) Events for Light-Water-Cooled Nuclear Power Plants.”
- Generic Letter 85-06, “Quality Assurance Guidance for ATWS Equipment That Is Not Safety-Related,” April 16, 1985 (Accession No. ML031140390).
- IEEE Std 7-4.3.2-2003, “IEEE Standard Criteria for Digital Computers in Safety Systems of Nuclear Power Generating Stations,”
- NUREG/CR-6463, “Review Guidelines on Software Languages for Use in Nuclear Power Plant Safety Systems”, June 1996

D3 Guidelines in Other Industries

- FAA: RTCA (Radio Technical Commission for Aeronautics) DO-178B Software Considerations in Airborne Systems and Equipment Certification
- DOD: MIL-STD-882C System Safety Program Requirements
- FDA: Review Guidance for Computer Controlled Medical Devices Undergoing 510(k) Review