Adaptive System Infrastructure for Ultra-Large-Scale Systems

SMART Conference, Thursday, March 6th, 2008

Dr. Douglas C. Schmidt

d.schmidt@vanderbilt.edu

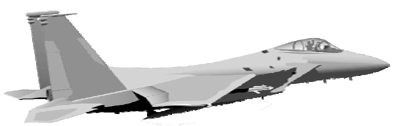
www.dre.vanderbilt.edu/~schmidt

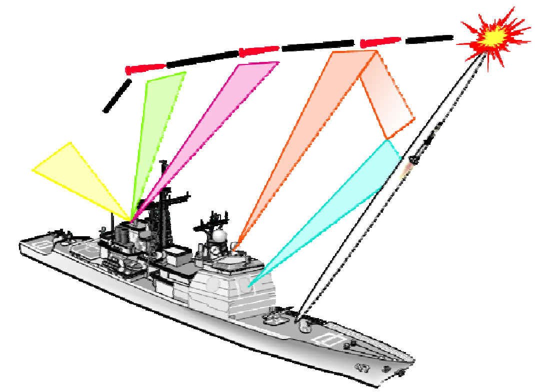
Institute for Software Integrated Systems

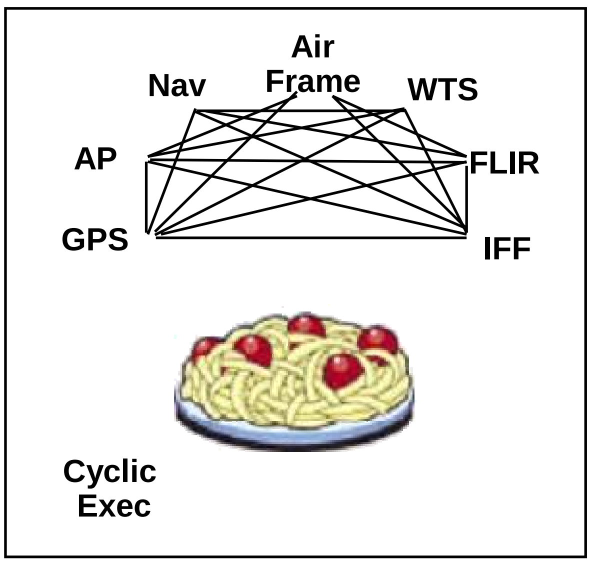
Vanderbilt University Nashville, Tennessee

Past R&D Successes: Platform-centric Systems

From this design paradigm…







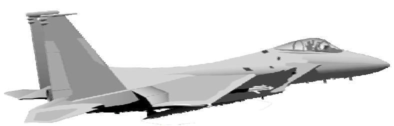
Legacy systems are designed to be:

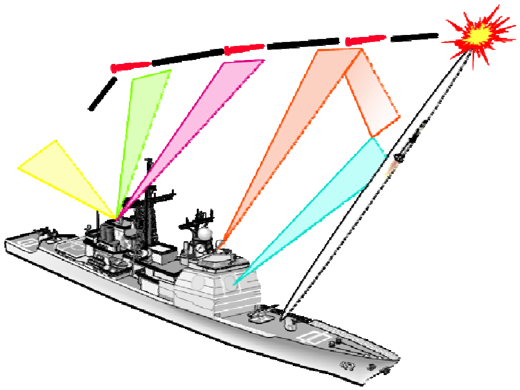
- Stovepiped
- Proprietary
- Tightly-coupled, brittle, & non-adaptive
- Expensive to develop & evolve
- Vulnerable

Problem: Small changes can (& do) break nearly anything & everything

Past R&D Successes: Platform-centric Systems

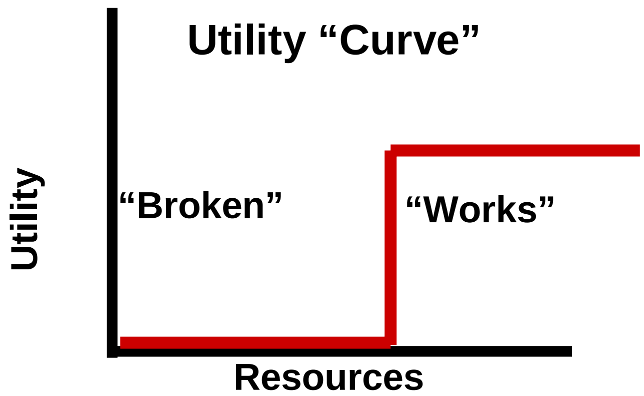
…and this operation paradigm…





Real-time quality of service (QoS) requirements for *platform-centric* systems:

- Ensure end-to-end QoS, e.g.,
  - Minimize latency, jitter, & footprint
  - Bound priority inversions
- Allocate & manage resources statically

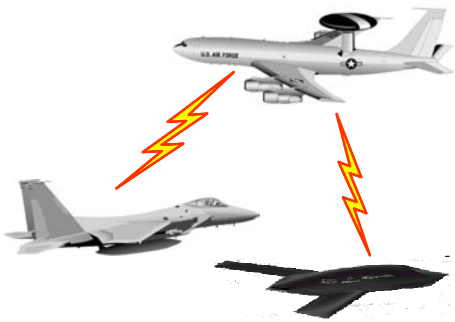


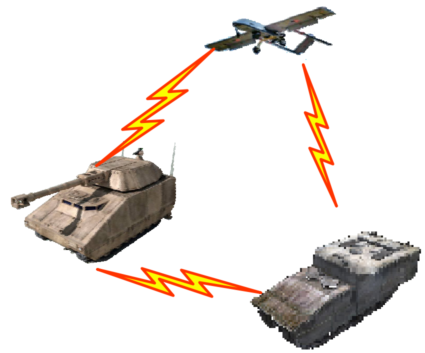
"Harder" Requirements

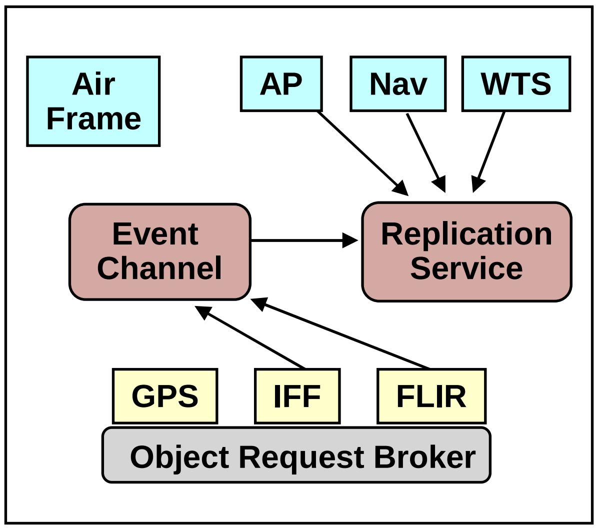
Problem: Lack of any resource can (& do) break nearly anything & everything

Past R&D Successes: Network-centric Systems

…to this design paradigm…





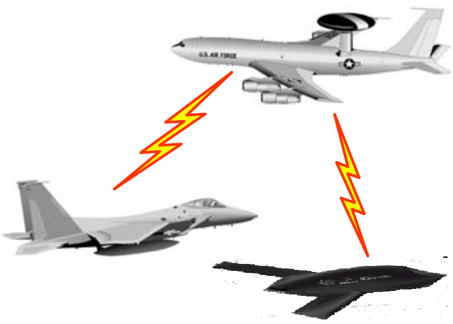


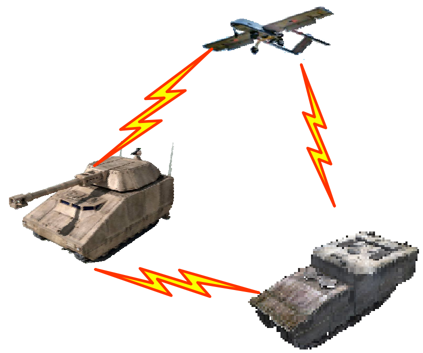
Today's leading-edge systems are designed to be:

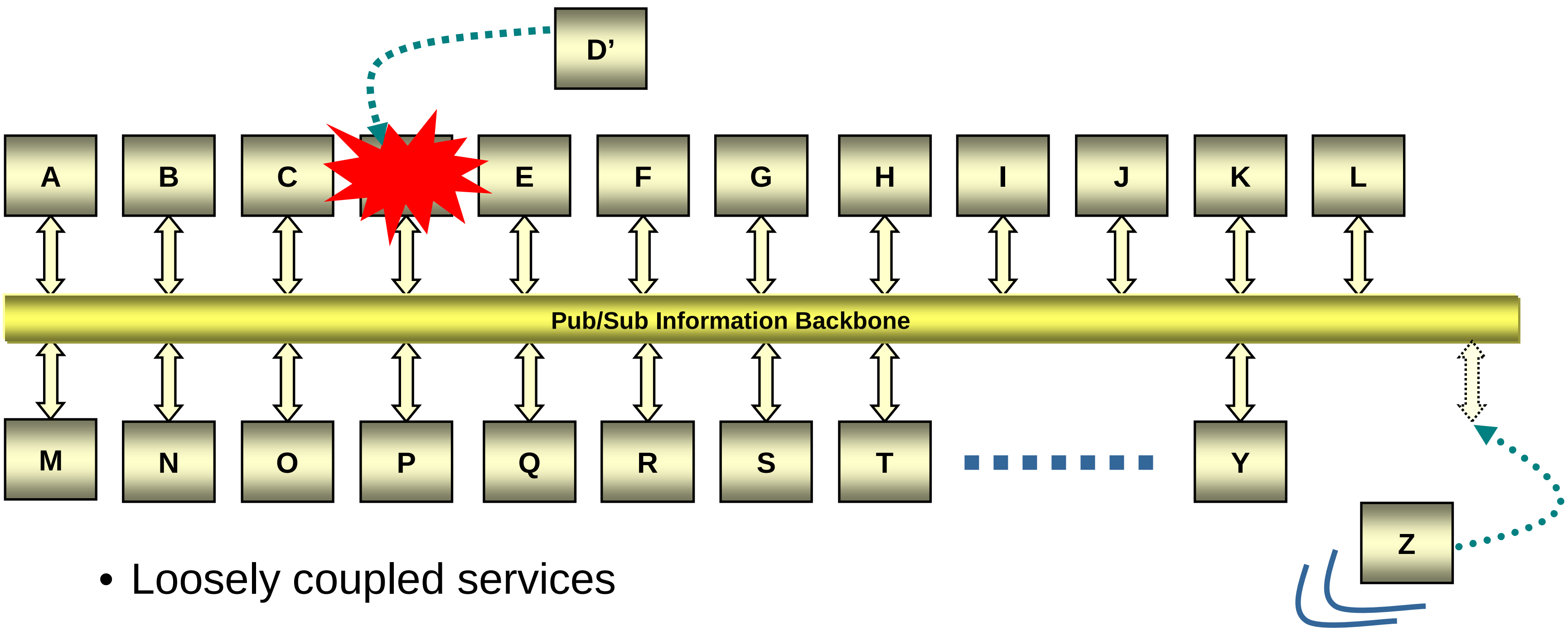
- Layered, componentized, & service-oriented
- More standard & COTS
- Robust to expected failures & adaptive for non-critical tasks
- Less expensive to evolve & retarget

Past R&D Successes: Network-centric Systems

…and this operational paradigm…



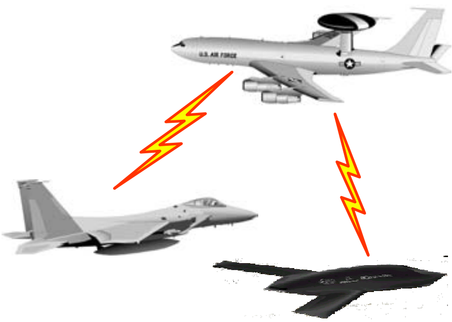


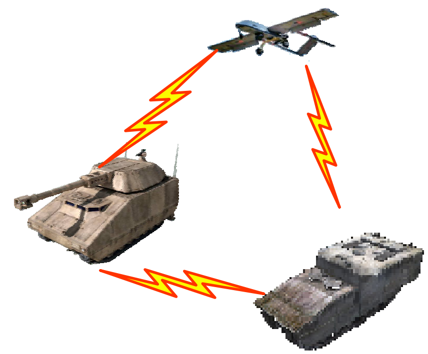



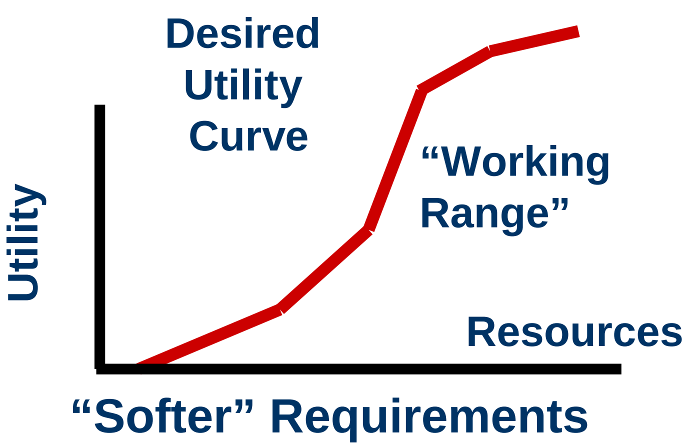
- Loosely coupled services

Past R&D Successes: Network-centric Systems

…and this operational paradigm…

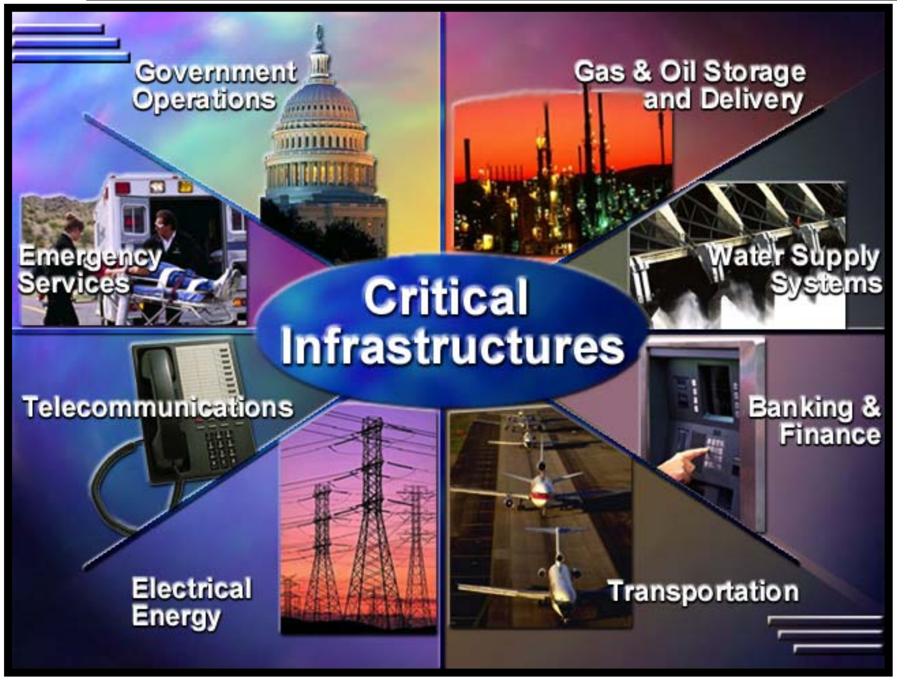






Problem: Network-centricity is an afterthought in today’s systems

System Infrastructure Demands in ULS Systems

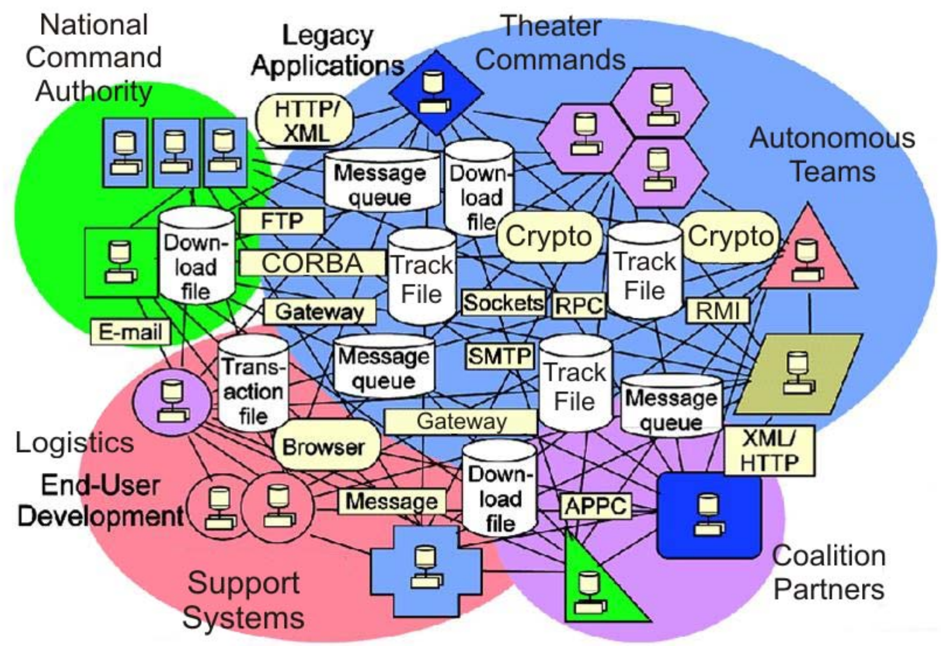


Key challenges in the ***problem space***

- Network-centric, dynamic, ultra-large-scale "systems of systems"
- Stringent simultaneous quality of service (QoS) demands
- Highly diverse & complex problem domains

Key challenges in the ***solution space***

- Enormous accidental & inherent complexities
- Continuous evolution & change
- Highly heterogeneous platform, language, & tool environments

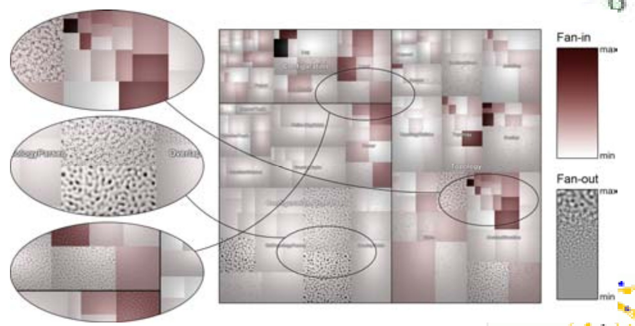


Conventional technologies ill-suited to meet ULS system infrastructure demands

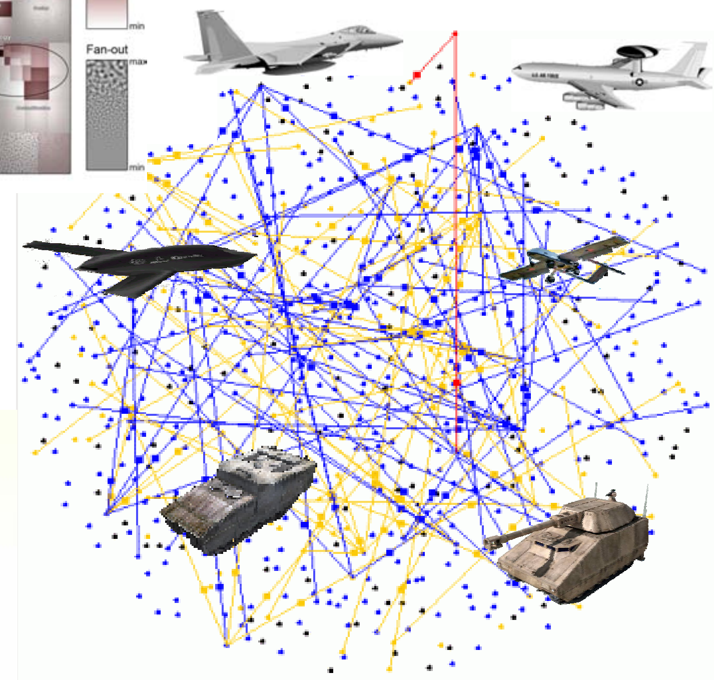
Promising R&D Areas for Adaptive ULS System Infrastructure

• Decentralized Production Management

• View-Based Evolution



• Evolutionary Configuration & Deployment

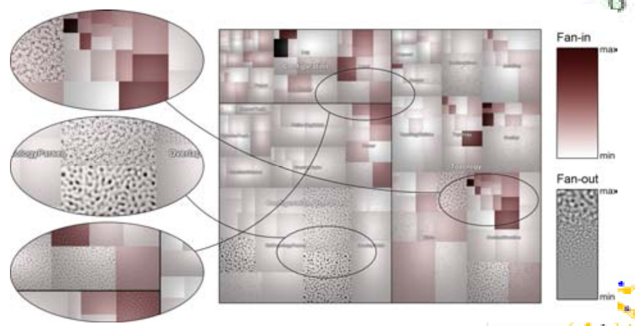


• In Situ Control & Adaptation

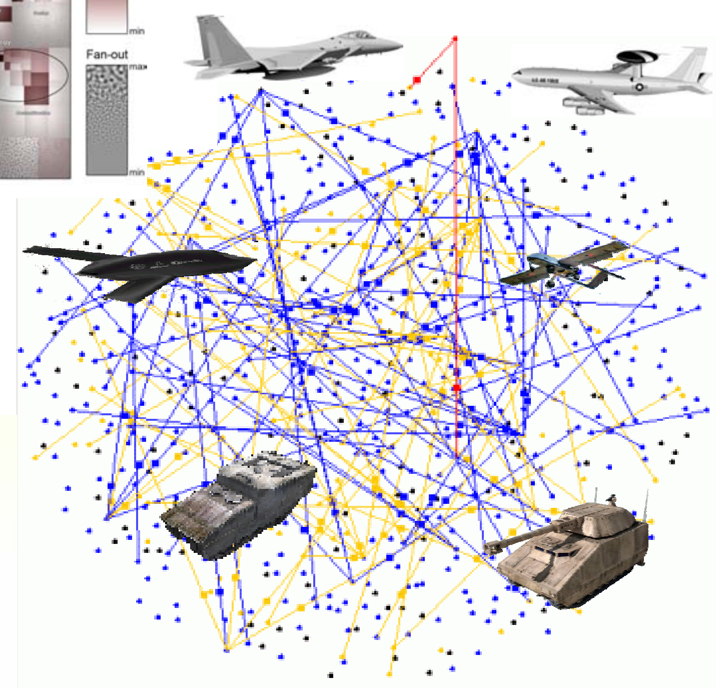
Promising R&D Areas for Adaptive ULS System Infrastructure

- Decentralized Production Management

- View-Based Evolution



- *Evolutionary Configuration & Deployment*

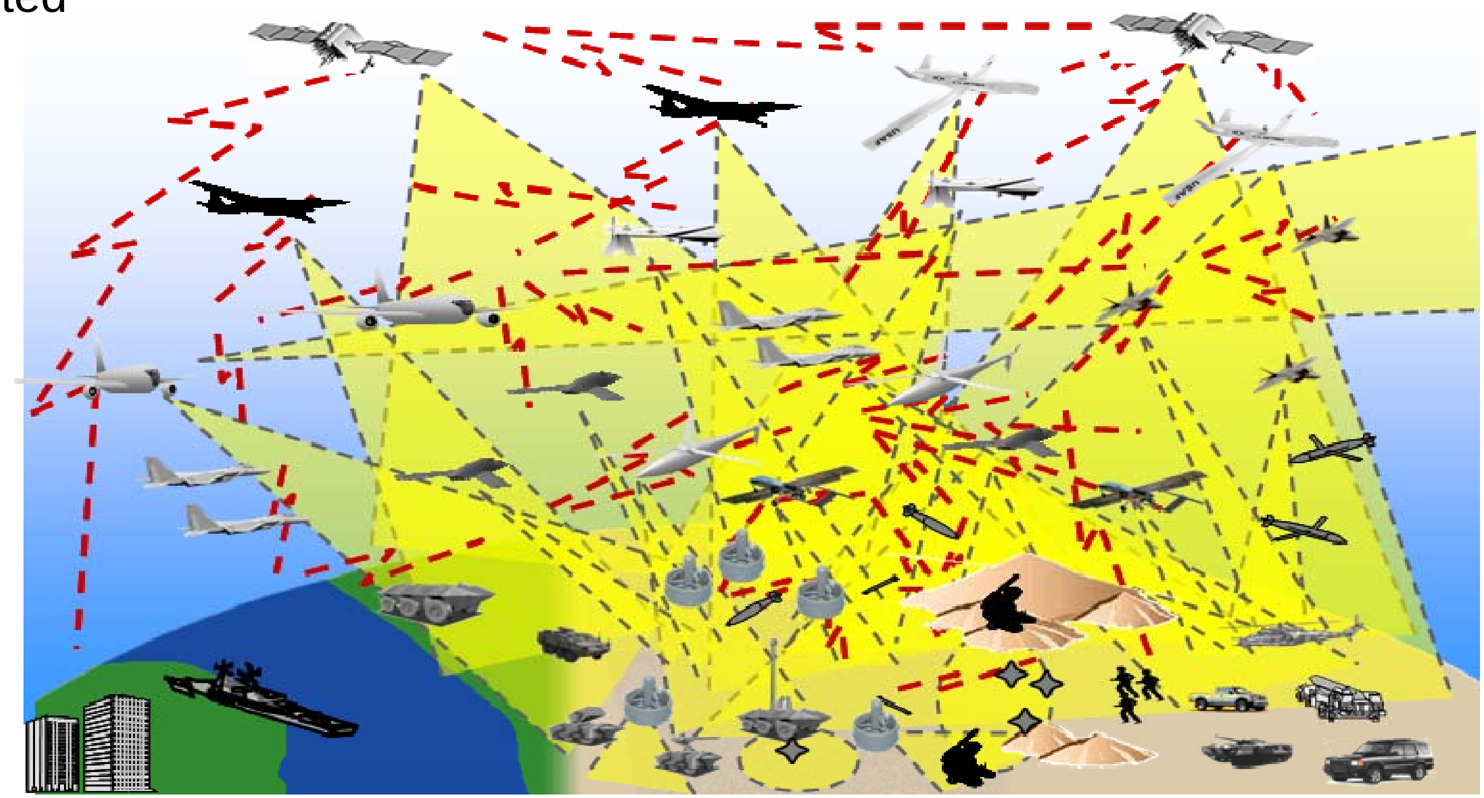


- *In Situ Control & Adaptation*

Evolutionary Configuration & Deployment

Goals

- Develop theory & concepts for ULS system configuration & deployment to distribute, customize, & install software components dependably & securely:
  - Despite an evolving mixture of proven & unproven components
  - Despite the existence of different versions of components in various deployment configurations
  - While providing the ability to rollback to proven configurations when problems are detected

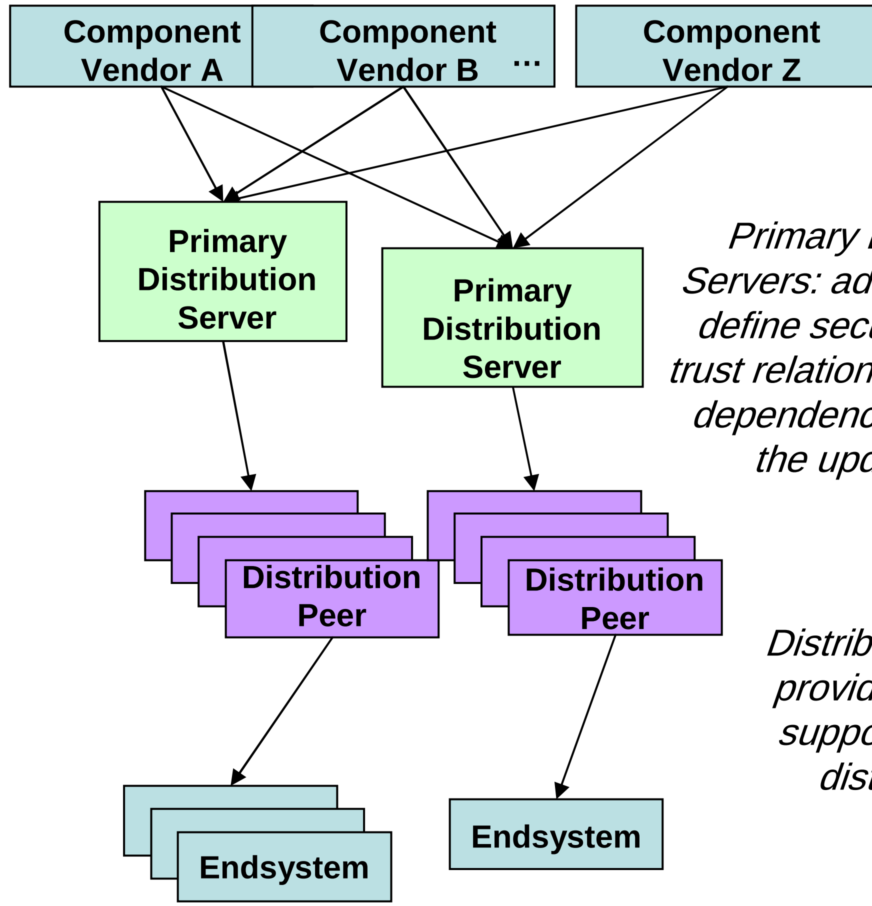


Evolutionary Configuration & Deployment

Promising Research Approaches

- Models, algorithms, & tools for specifying, reasoning about, & modifying ULS system components dependencies to validate key functional properties
- System execution modeling techniques & tools to analyze & optimize system QoS before & during software updates
- Scalable protocols for automatically distributing software updates dependably & securely under hazardous operating conditions

Component Vendors: Supply components



Primary Distribution Servers: add metadata to define security policies, trust relationships & critical dependencies, & initiate the update cycle

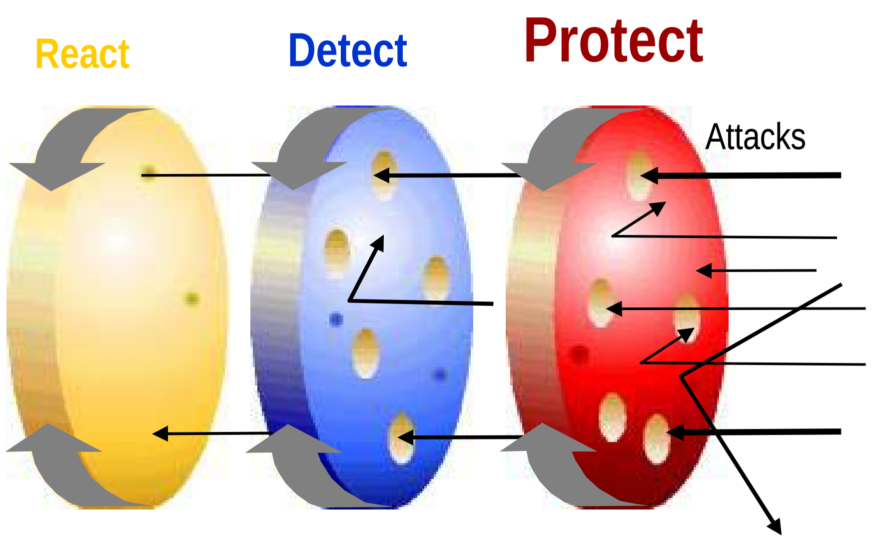
Distribution Peer: provide scalable support for ULS distribution

Endsystems: provide mechanisms to support component update lifecycle (download, verify, activate, monitor, fallback, report etc.)

In Situ Control & Adaptation

Goals

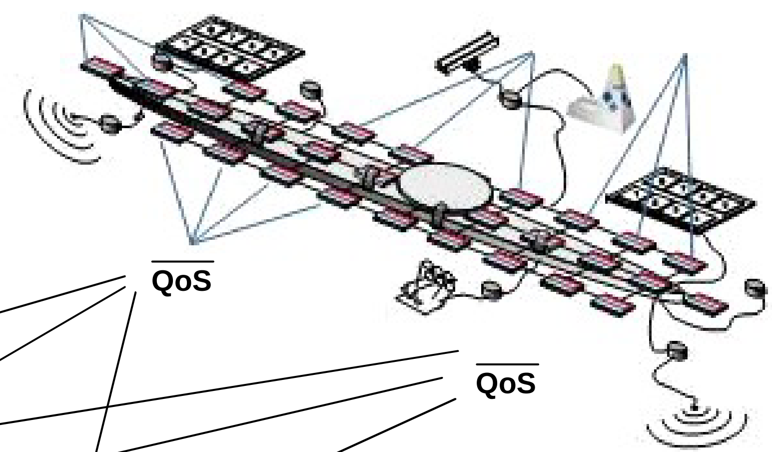
- Develop theories, algorithms, & services that allow ULS systems to
  - Monitor the activity of system elements & their environments
  - Perform self-testing to detect deviations in expected behavior & performance & automatically recover from them
    - e.g., by reconfiguring component behavior & configurations while the system is operating
  - Protect the system from damage when patches & updates are installed, as well as from attacks perpetrated against them during operation

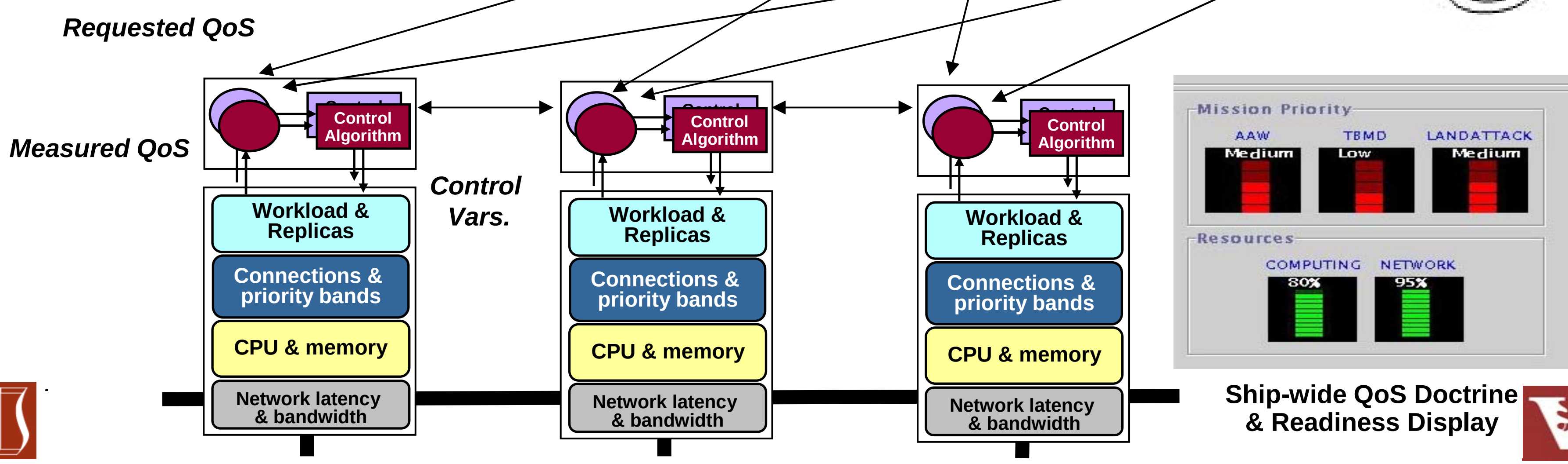


In Situ Control & Adaptation

Promising Research Approaches

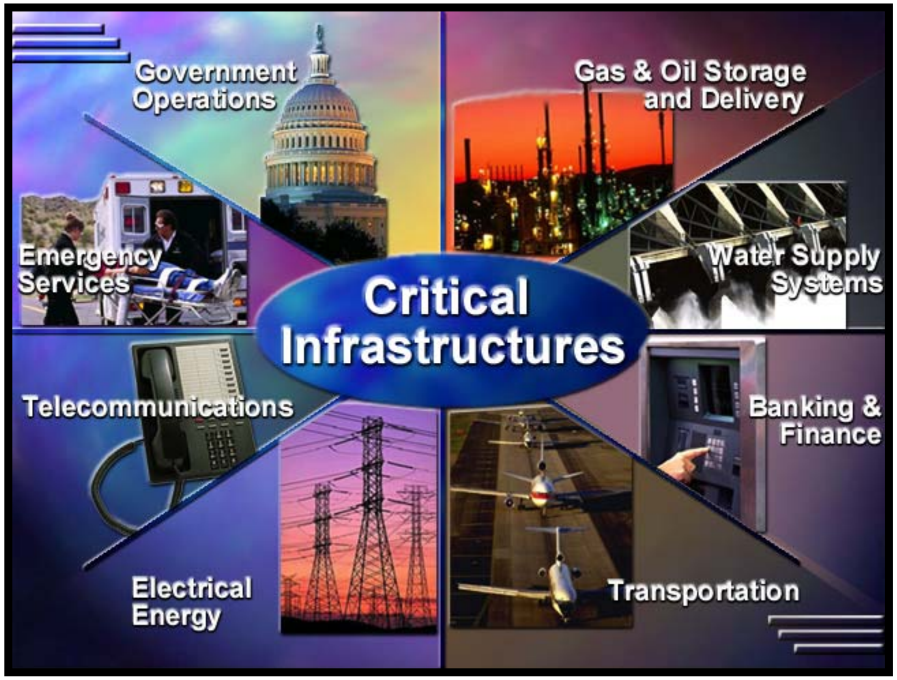
- Control-theoretic techniques that handle rapidly changing demands & resource-availability profiles & configure these mechanisms with service policies tuned for different operating modes
- Scalable techniques for developing controllers that adapt ULS systems under a wide range of conditions
- Certification techniques & processes that can ensure adaptive systems only operate within safe, correct, & stable configurations





Concluding Remarks

- The emergence of ULS systems requires significant innovations & advances in adaptive system infrastructure
- Not all technologies will provide the precision we're accustomed to in legacy small-scale systems
- Breakthroughs in computing technology & related disciplines needed to address ULS system infrastructure challenges
- Initial groundwork layed in various R&D programs



PCES

MoBIES

ARMS

NEST

AFRL S²PRUCE²
Systems and Software Producibility Collaboration and Evaluation Environment

Much more research needed on adaptive infrastructure for ULS systems