COMMAND AND CONTROL TECHNOLOGIES
CORPORATION

The Role of Architecture in Managing COTS Based High Integrity Systems

Ground Systems Architecture Workshop

March 2003

Contact Information

Rodney Davis

Chief Technology Officer

(321) 264-1193

davisrd@cctcorp.com

http://www.cctcorp.com

COMMAND AND CONTROL TECHNOLOGIES CORPORATION

Agenda

- Strategic Questions for COTS Based Systems
- What is a High Integrity System?
- What is Architecture?
- What is COTS?
- What is the Issue with COTS?
- So Why Use COTS?
- System Properties for COTS Based Systems
- Key Architecture Principles
- Architectural Properties for COTS Based Systems
- Strategies Summary
- Examples
- References
- Summary and Conclusions
- Backup

Strategic Questions for COTS Based Systems

http://www.sei.cmu.edu/cbs/

- Which technologies and products are most appropriate?
- How can product mismatches be rectified?
- How can we engineer system attributes such as reliability, security, and performance in spite of decreasing control over individual system components?
- How do we integrate COTS products with the custom code that continues to provide the core of many systems?
- How do we take advantage of COTS while delivering a system that can evolve over a long lifetime?

Architecture strategies can help address these questions!

What is a High Integrity System?

- Software that must be trusted to work dependably in some critical function, and whose failure to do so may have catastrophic results, such as serious injury, loss of life or property, business failure or breach of security.
- 3 Main Objectives of High Integrity Systems:
  - Confidentiality – Protecting against unauthorized accidental disclosures of info caused by system failures or user error.
  - Integrity – Protecting against unauthorized modification or info, protecting against unintentional mod of info caused by system failures or user errors
  - Availability – Protecting against unauthorized withholding of info or resources, protecting against failure of resources

What is Architecture?

- Architectures deal with abstraction, decomposition and composition, style, and aesthetics. Management of systems architecture provides the means for controlling complexity
- Deals with the design and implementation at the highest level. Postponing the detailed decisions until the architecture foundations are laid is critical
- Architecture is the set of decisions about any system that keeps its implementers and maintainers from exercising needless creativity
- The architecture of a system consists of the structure(s) of its parts, the nature and relevant externally visible properties of those parts, and the relationships and constraints between them

What is COTS?

- **FAR** - Something that one can buy, ready-made, from a manufacturer.
  - It exists a priori
  - Its available to the general public
  - It can be bought, leased, or licensed.
- **Other Variants**
  - Customer has no control over specification, schedule, or evolution
  - Supported and evolved by the vendor, who retains IP rights
  - Used without source code modification
- Further Extension
  - Open source software can be viewed as COTS in many regards

What's the Issues with COTS?

- Loss of control
  - Schedule, license agreements, & product discontinuation
  - Will the COTS vendor be in business in a few years?
  - The external functionality of a COTS product is defined by the vendor.
- Introduces discontinuity in the understanding of the system as a whole.
  - Design information is typically limited.
  - They are often provided as "black boxes".
  - There is usually little revelation of the rationale underlying the product design
  - Deficiency of info increases the possibility of introducing design errors
- COTS don't necessarily conform to interface standards
  - Lack of commonality with other products
  - Long-term maintenance issues, especially for long life cycle systems
- Complexity becomes more difficult with multiple COTS integration, particular with regard to maintenance
- Perceived benefits often center around claims of "tried and tested".
  - Claims are rarely expressed in a form that can be applied to address regulations.
- Strict compliance with regulations/standards may exclude the use of COTS in critical systems, especially in higher integrity systems.

So Why Use COTS?

- Cost savings and avoidance
  - You don't have to pay for the capabilities improvements driven from the market
    - Continuous change and improvements are a two edged sword
  - Cost of maintenance is amortized over many customers
- Increased capability
  - Usually a faster life cycle
- Increased maturity … hopefully
- Can't get there from here without it

System Properties for COTS Based Systems

- Service Level Tuning – Service tuning should be easy
- Adaptable Configuration – Components are added, deleted, updated, or replaced over time.
- Visibility – Need to be able to monitor COTS behavior
- Fault Isolation and Exception Handling – Identify and isolate faults ASAP.
- Open System – Openness allows for extension and integration
- Security – Enabled by architecture
- Integration Support – Architecture defines and supports integration (exception handling, concurrency control, common data models, & common user interface)

COMMAND AND CONTROL TECHNOLOGIES CORPORATION

Key Architecture Principles

- Abstraction / Simplicity- Most important quality
- Interoperability – Ability to change functionality and interpretable data between entities
- Extensibility – Support for unforeseen uses & new requirements. Sometimes in conflict with Interoperability
- Symmetry – common interface for a wide range of components
- Separation of Concerns – Limits scope of changes as system evolves
- Metadata – Support for dynamic reconfigurability via self-descriptive services/information
- Separation of Hierarchies - Layering

Architectural Properties for COTS Based Systems

Architecture approaches to achieve desired properties

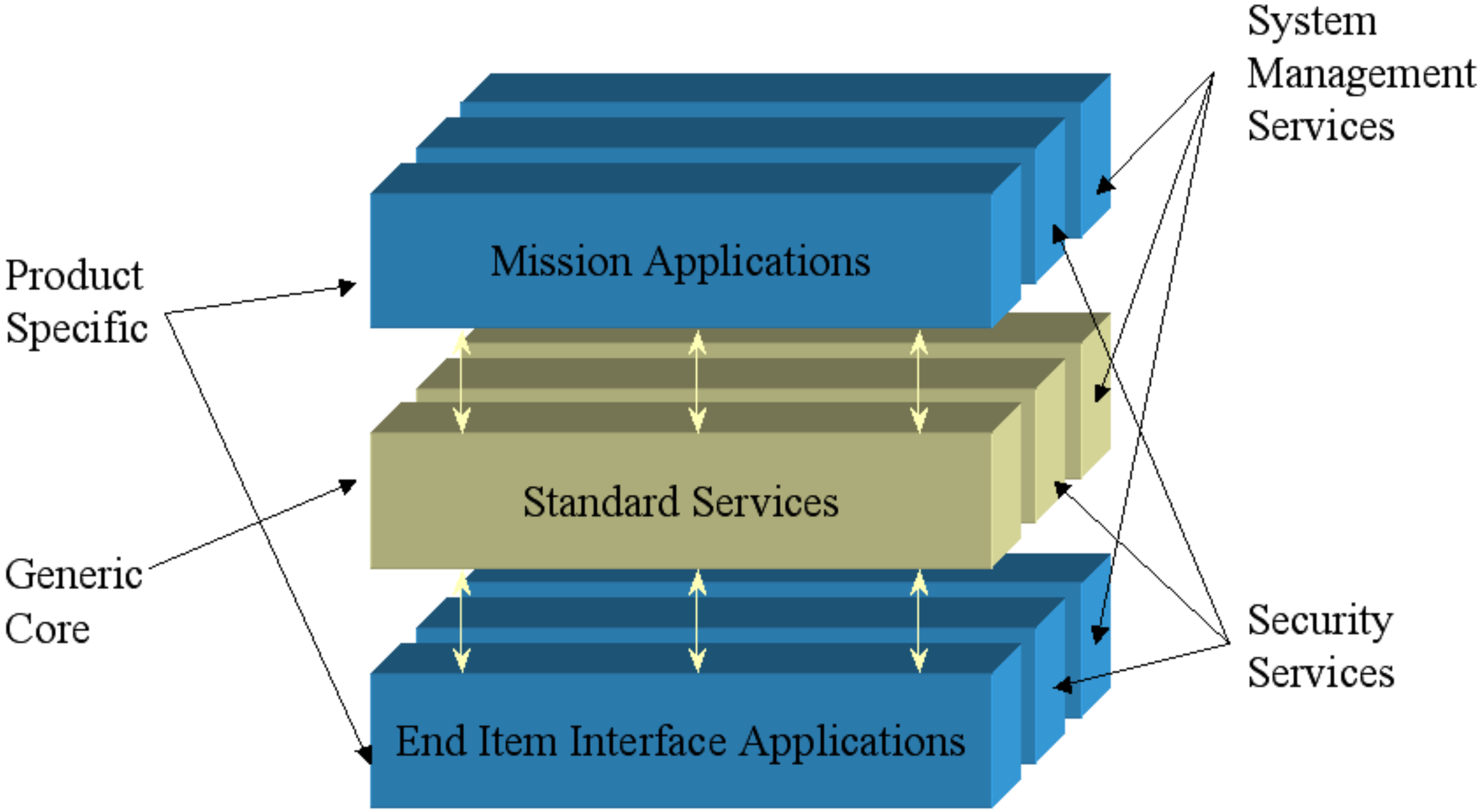
- **Connection Infrastructure** – means of info and control transfer
  - Minimize coupling, & emphasize adaptability
  - Minimize connection types (Pipes, procedure calls, events, etc.)
  - Use open standards
  - Consider security
- **Interconnection Topology** – Minimize number of interconnections and increase understanding (KISS)
- **Interfaces** – defines syntax and semantics.
  - Interface standards – only available in mature domains
  - Wrappers/adapters
    - To conform to Standards
    - To reduce impact to system for changes to a wrapped comonent
    - Provide a standard interface
    - Add or hide functionality
    - Control look and feel
    - Improve security level
- **Tailorability** – Allow for adaptability without accessing source
- **Architectural Style** – Interconnection strategy of components and control and data flow (I.e. Multi-tiered client/server, Dataflow, Mediator)
- **Run-time Instrumentation** – Both in COTS and glue for fault isolation, & troubleshooting
- **Collaborations** – Simplifies complex, multi-component services.

Strategies Summary

- Choosing Wisely
  - Technology and product forecasting
    - Get involved in the market. Apply influence when you can
  - Open Systems/Standards
    - Can offset issues rapid technology change
    - Open specs, interfaces, services, and data formats (Fully defined, available to the public, maintained by consensus (market vs. formal?)
    - Avoid becoming captive to the products of a single vendor
- Product evaluation and qualification
  - Focus is to build understanding/experience
  - Look out for systematic errors arising from badly fitted COTS
- Architect/Engineer for security and reliability
  - Use of patterns can be helpful
    - Facades and adapters for isolation
    - Mediators to encapsulate interdependencies between resources
  - Patterns may have to be mined from COTS during evaluation
  - Must retain structure and cohesiveness yet allow the system to respond easily to changes
  - Analysis of alternatives
  - Document architecture implications of decisions. Supports evolving architecture

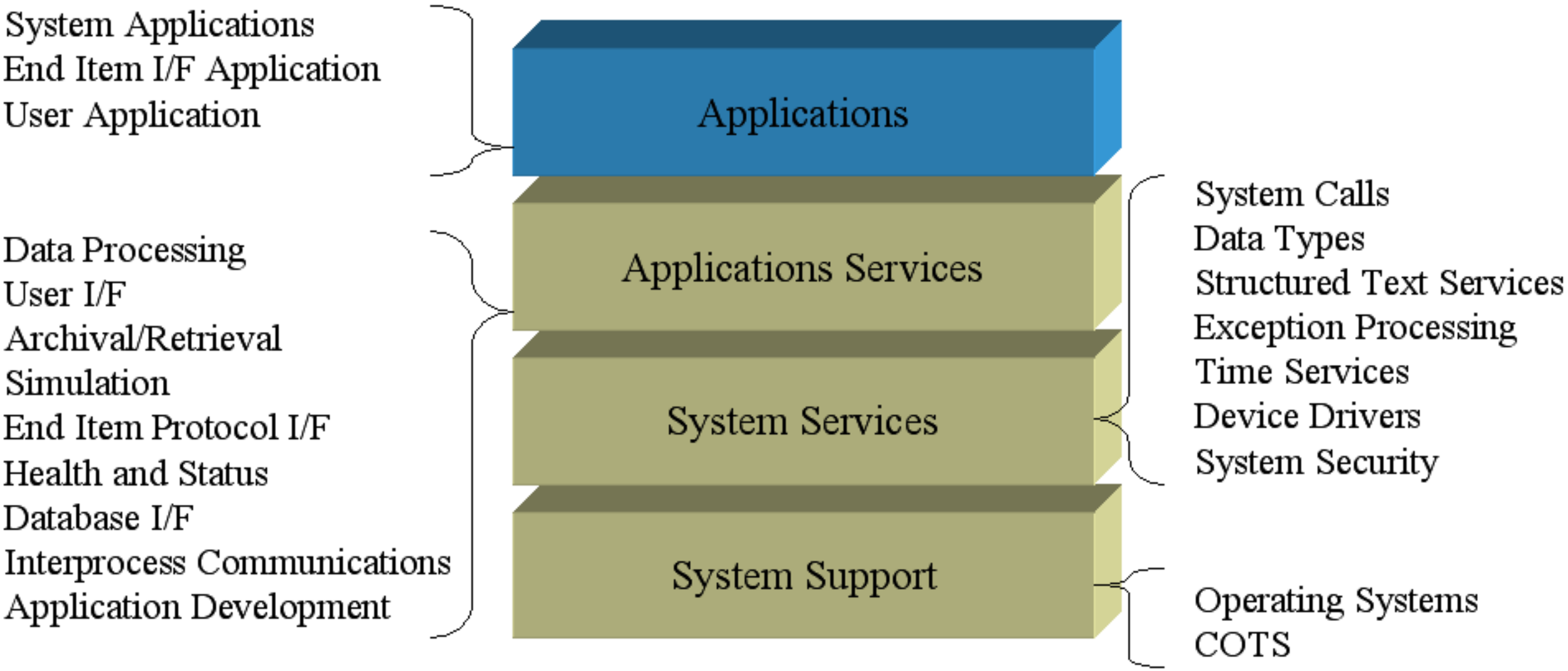
COMMAND AND CONTROL TECHNOLOGIES CORPORATION

Abstract Reference Model

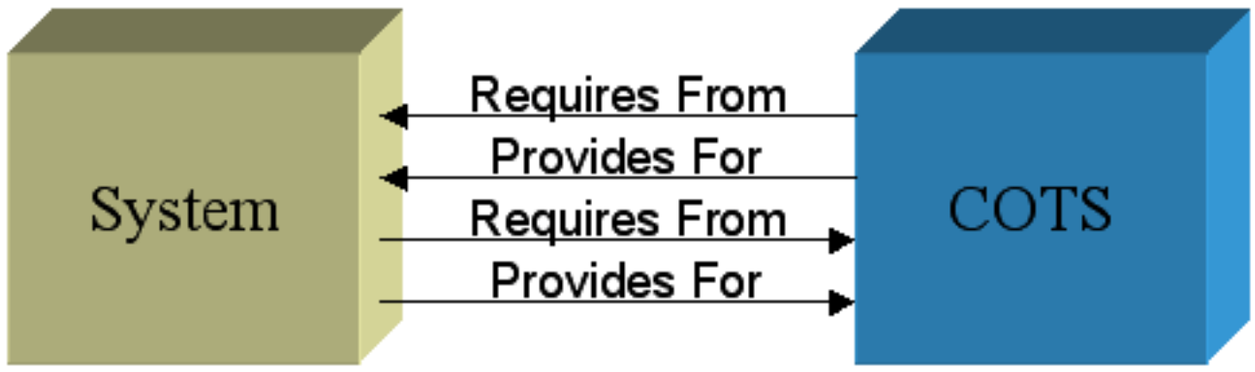


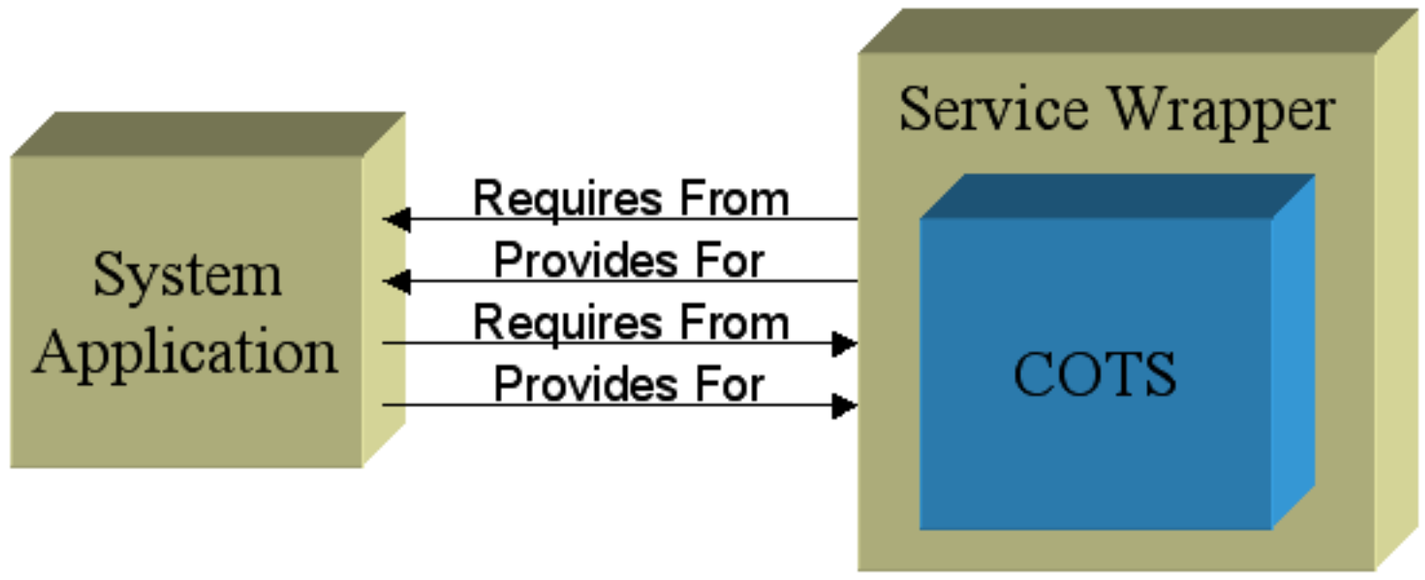
COMMAND AND CONTROL TECHNOLOGIES CORPORATION

Example Layered Architecture



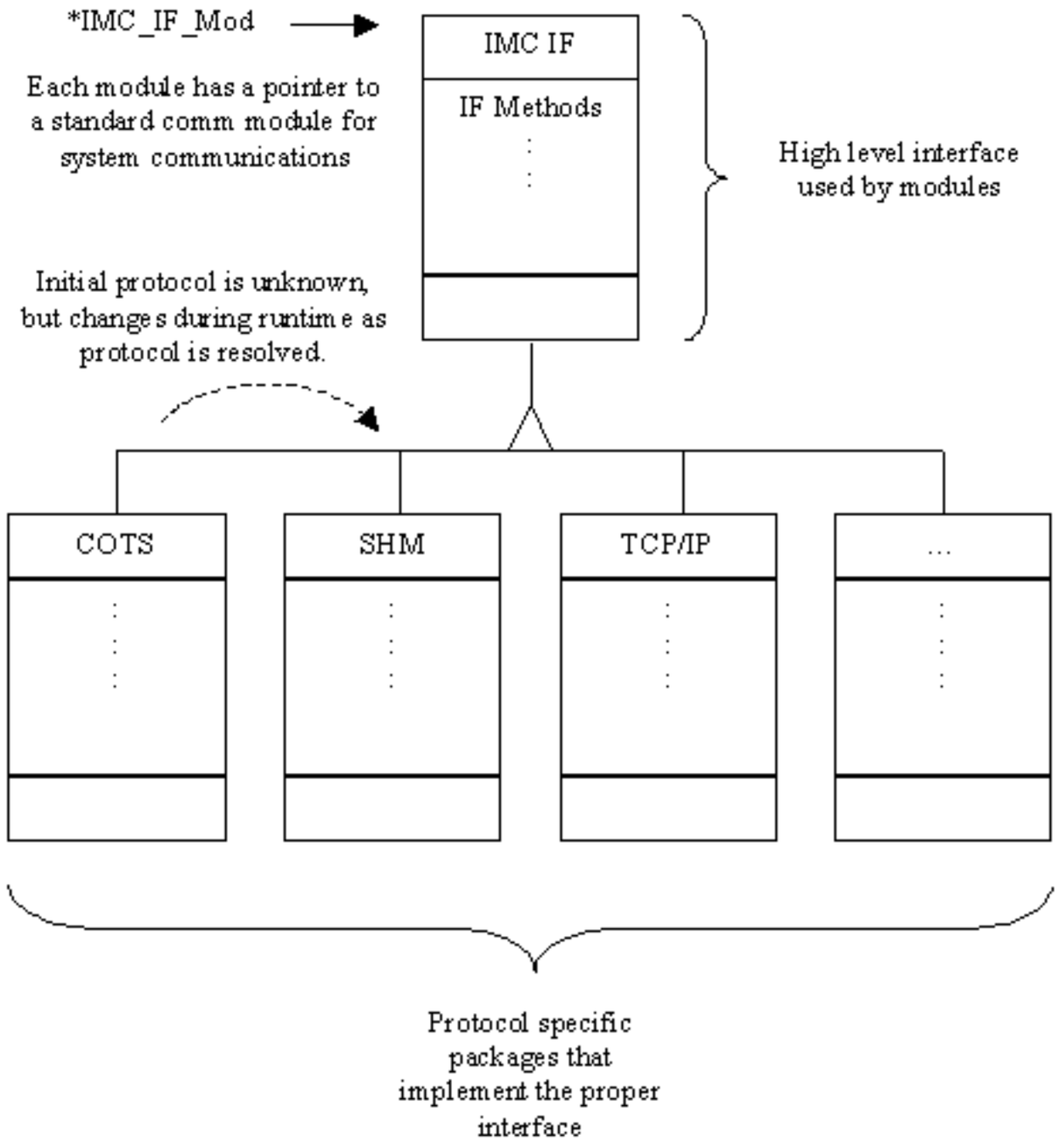
Relationships Between COTS and System





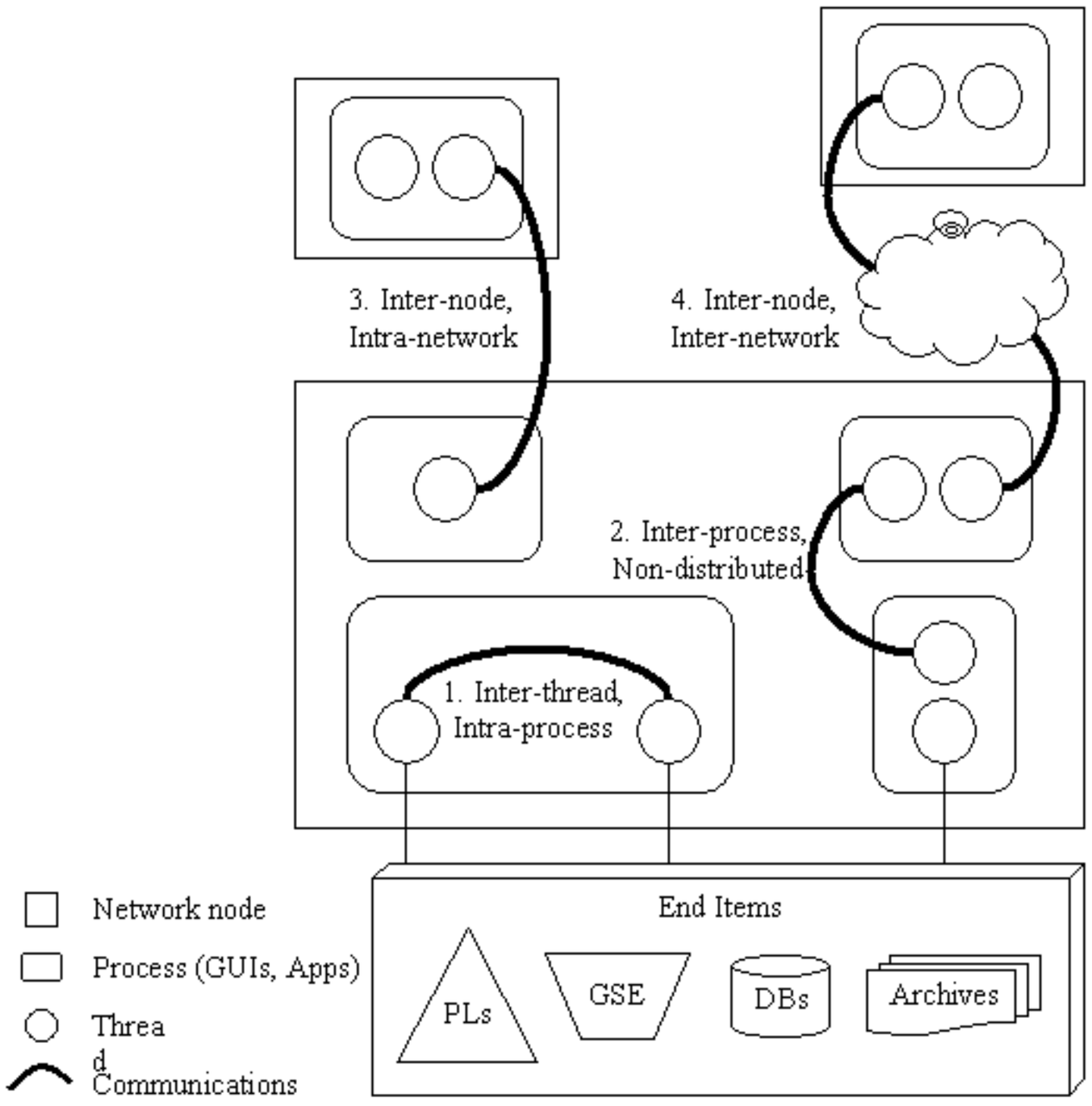
COMMAND AND CONTROL TECHNOLOGIES CORPORATION

Example Encapsulation Approach



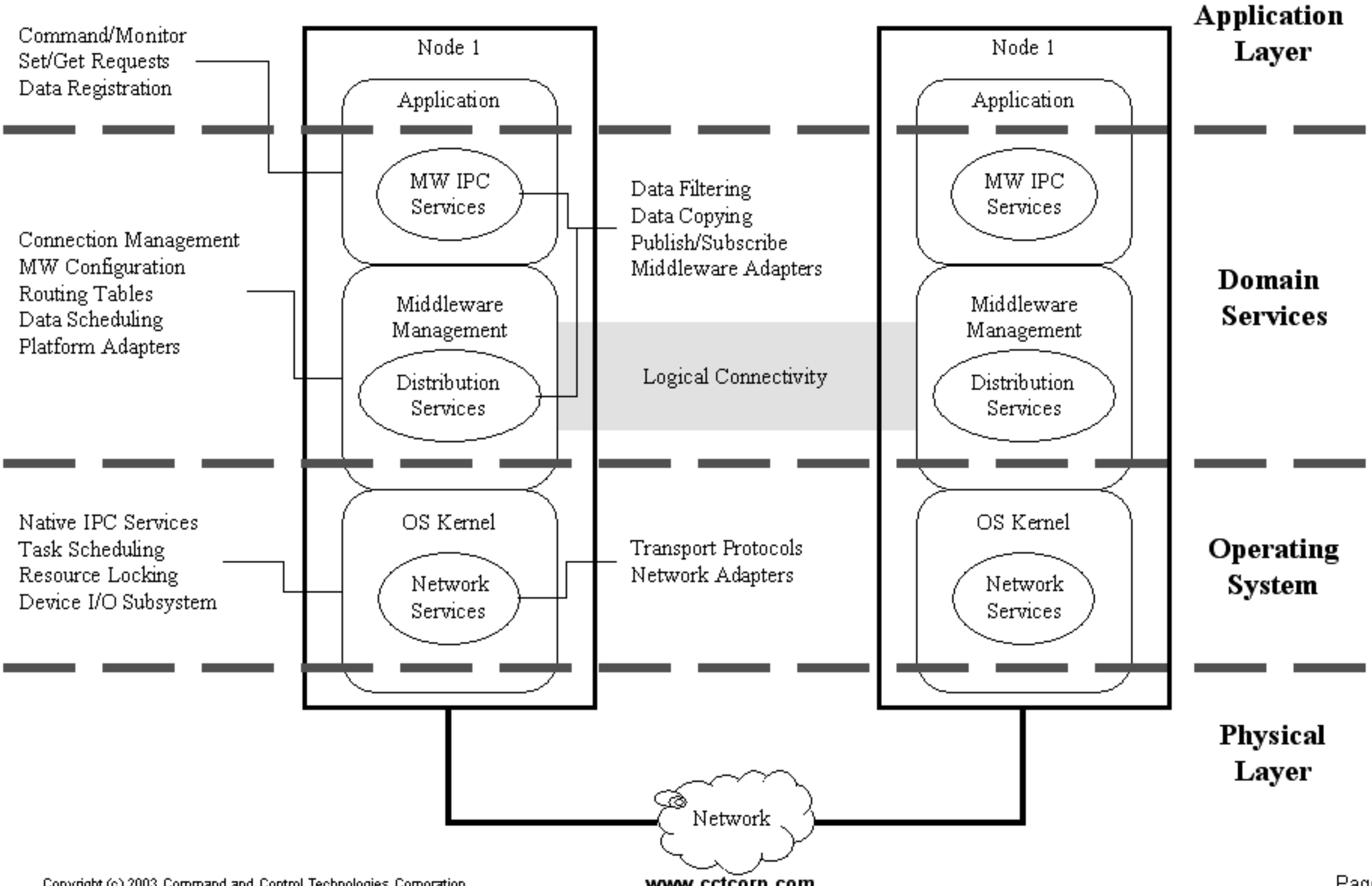
COMMAND AND CONTROL TECHNOLOGIES CORPORATION

Example Connection Infrastructure



COMMAND AND CONTROL TECHNOLOGIES CORPORATION

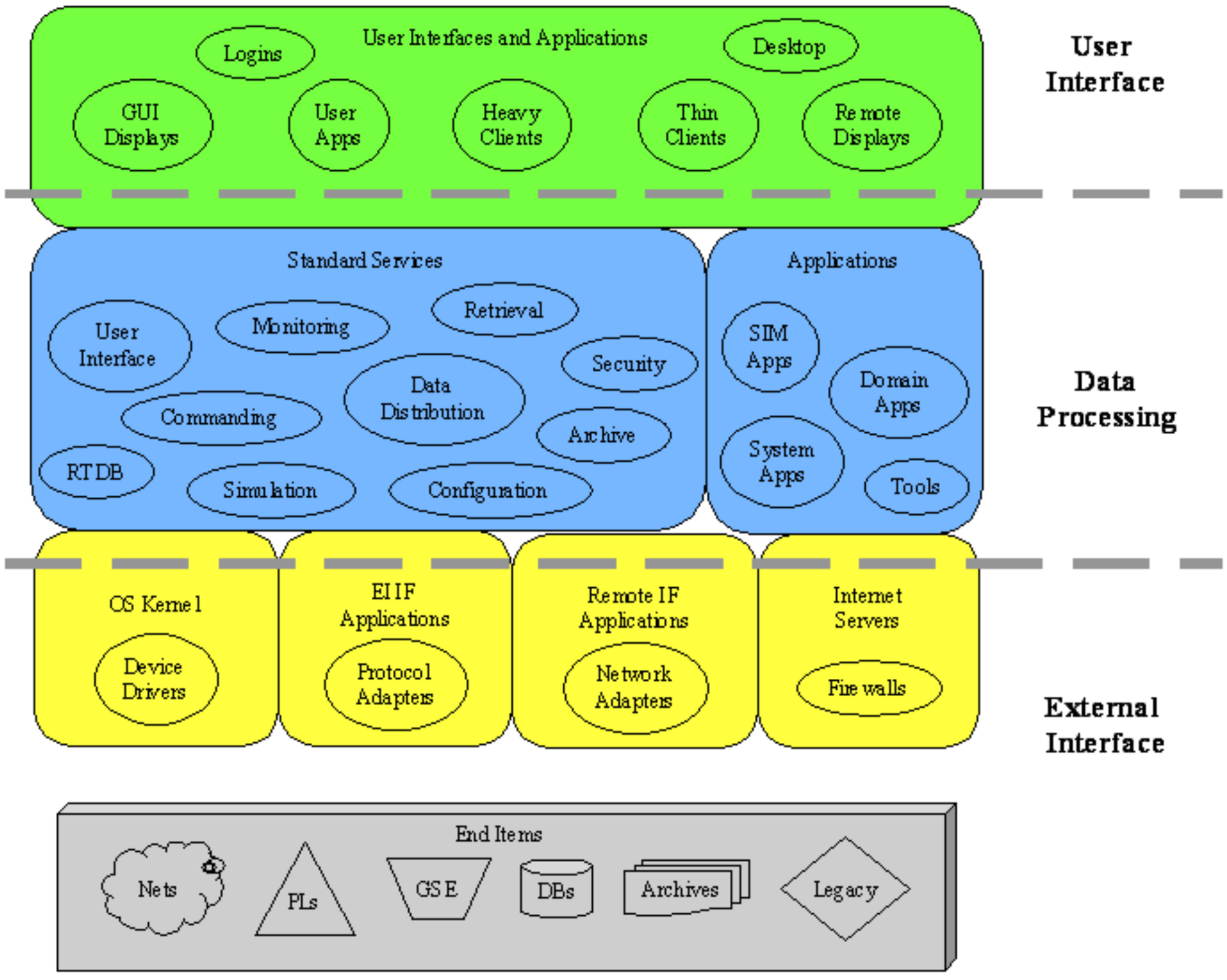
Example Service Layers



COMMAND AND CONTROL TECHNOLOGIES CORPORATION

Example Concepts

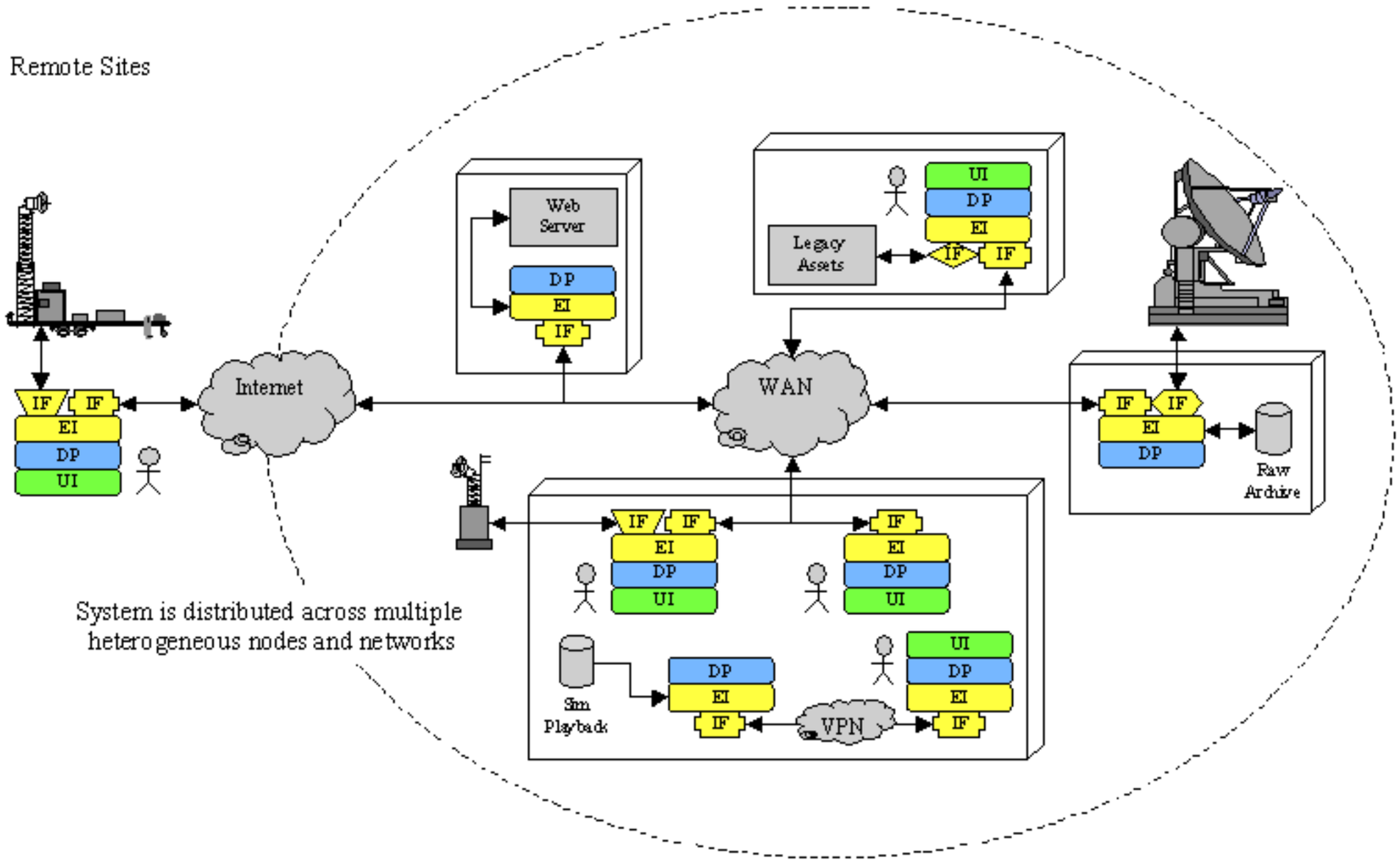
Common Classification Architecture



COMMAND AND CONTROL TECHNOLOGIES CORPORATION

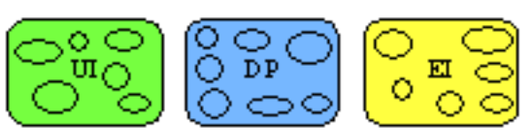
Example Concepts

Common Interoperable Architecture

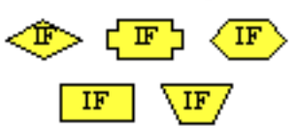


System is based on a common Reference Architecture

System is built from standard general-purpose components



System reuses special-purpose interface adapters



COMMAND AND CONTROL TECHNOLOGIES CORPORATION

Summary

- Architecture for COTS integration is crucial to system integrity and system life cycle
- Understanding of COTS is critical to a complete architecture
- Traditional strategies of good design practice are also applicable to COTS integration. Although scoping is necessarily more organic or continuous

Contact Information
Rodney Davis
Chief Technology Officer
(321) 264-1193
davisrd@cctcorp.com
http://www.cctcorp.com

Backup Slides

References

- IEC 61508 Standard
- Architecture-Centered Information Systems in the Manufacturing Domain, Alleman, 2002
- Designing High Integrity Systems using Aspects, Georg
- Supporting the use of COTS in Safety Critical Systems, Dawkins , Kelly, 1997
- COTS-Based System Initiative www.sei.cmu.edu/cbs/
- Evolutionary Process for Integrating COTS-Based Systems (EPIC), Albert, Brownsword, SEI, 2002
- A Summary of DOD COTS-Related Policies, Oberndorf, Carney, SEI, 1998
- An Architecture for COTS Based Software Systems, Vigder, 1998
- Definition and Classification of COTS, Morisio, Torchiano, 2002
- Open Source Software, The Other Commercial Software, Hissam, Weinstock

Relevant Gov COTS Policies

- Clinger-Cohen: “Increase acquisition and incorporation of COTS”.
- FAR: Acquire COTS & NDI when available to needs
- DODD 5000.1: “If use or mod of existing … equipment will not meet the need, give top priority to … COTS
- DOD 5000.2-R: “Consider COTS/NDI the primary source of supply”
- JTA & DII/COE: Specifies C4I performance standards info processing, transfer, content, format and security.
  - Emphasis is interoperability
  - DII/COE establishes compliance levels

www.sei.cmu.edu/cbs/monographs.html