Design Principles for Secure Smart Grid System Integration across Network perimeters

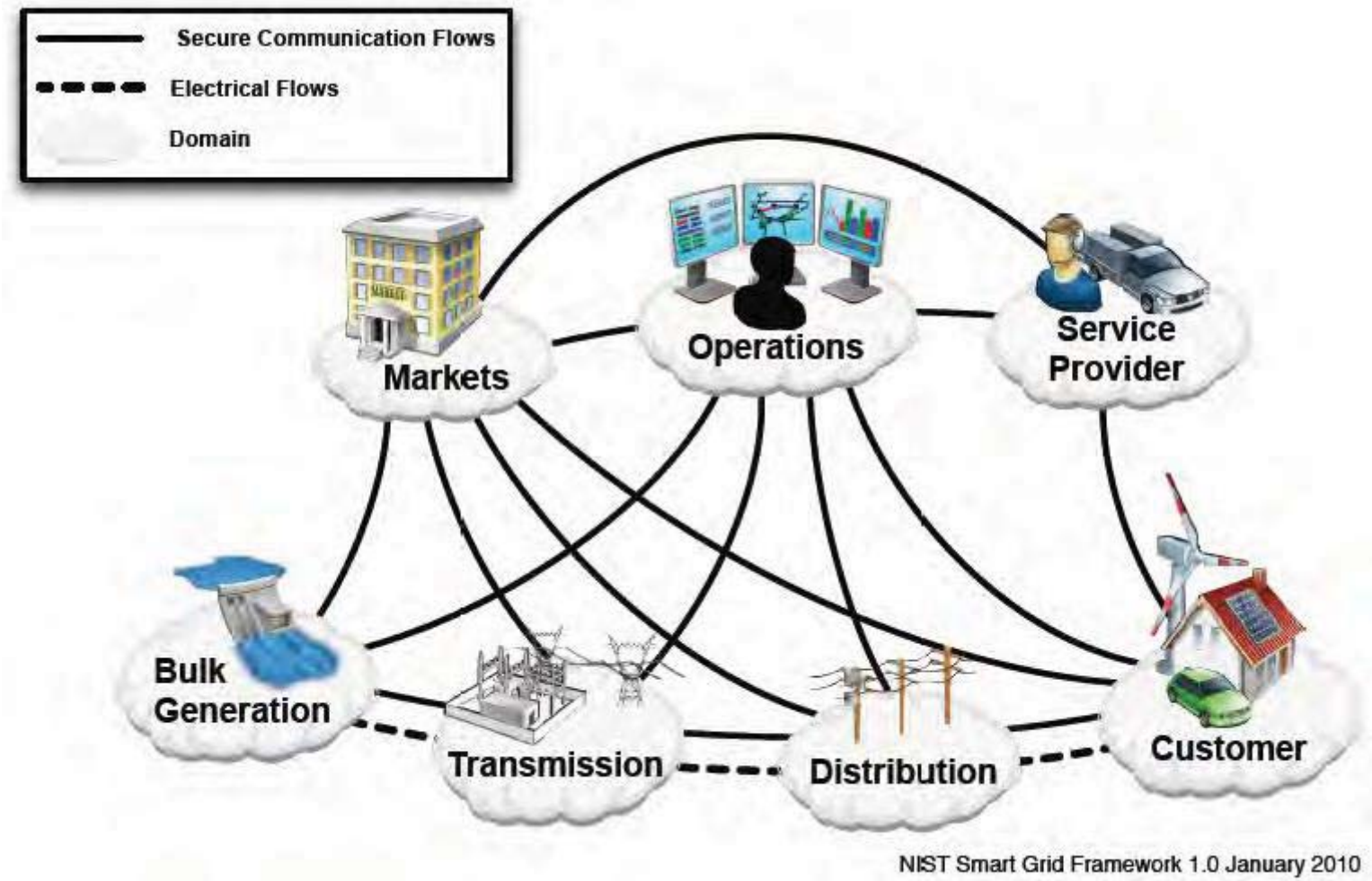
Sitaraman Lakshminarayanan
Security Architect – Systems Security
GE Energy

imagination at work

Overview

- Smart Grid System , Sub systems and System of Systems
- Software Applications and Smart Grid Domains
- Smart Grid System and Application Interfaces
- System Architecture (Service Oriented Architecture) for Integrating applications in various Smart Grid domains
- System behavior and Software Security
- Integrity and Non-repudiation and how it relates to Integrated Systems
- Impacting applications in Control center due to vulnerability elsewhere
- Understanding Software Security and Interoperability
- Threat analysis on Software Integration Layer. (integrity, Privilege escalation, etc.)
- Design principles for Software Security Architecture

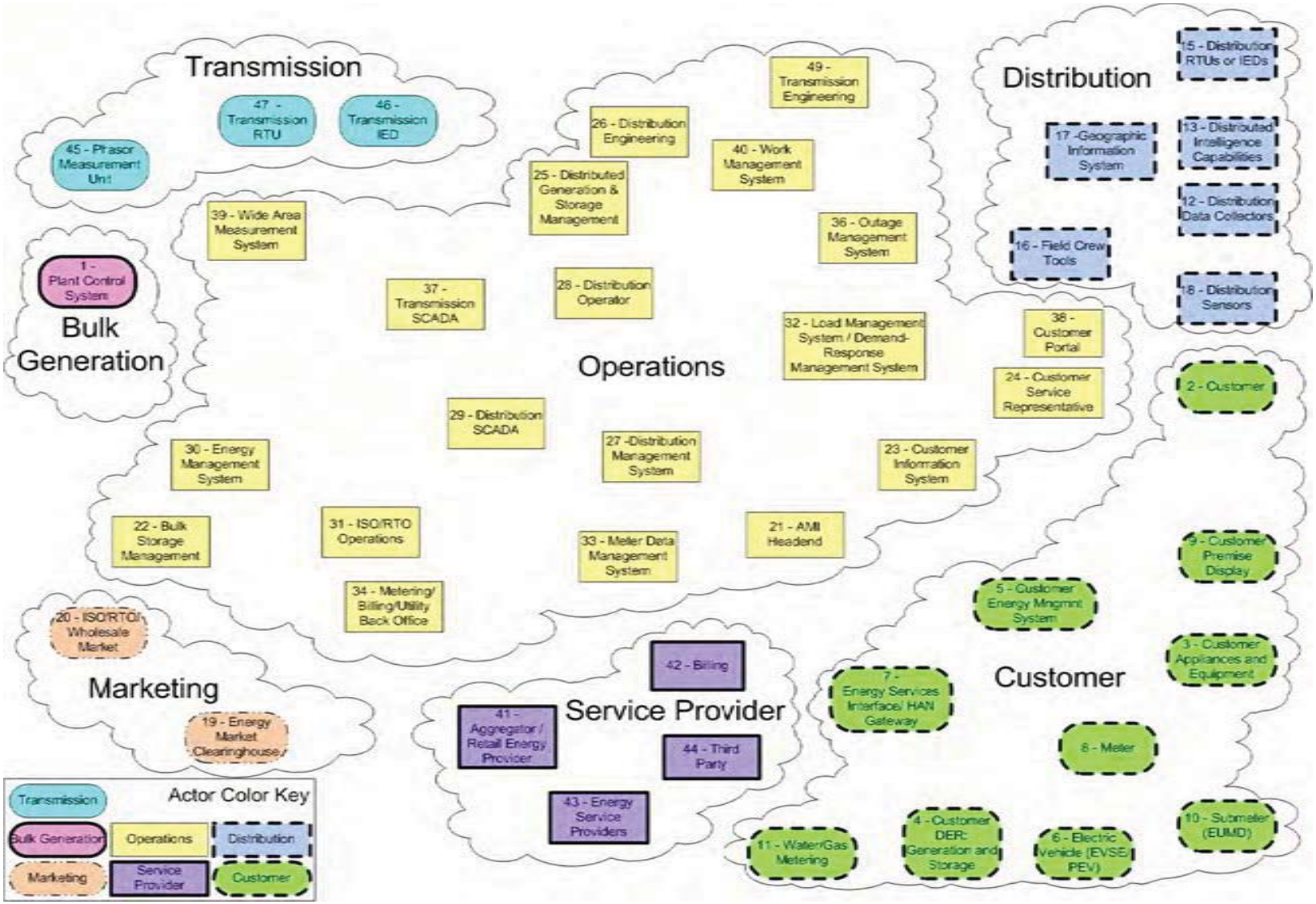
Smart Grid Domains : from NIST IR 7628



Smart Grid Domains : from NIST IR 7628

- Multiple Domains
- Each domain may contain application from different vendors

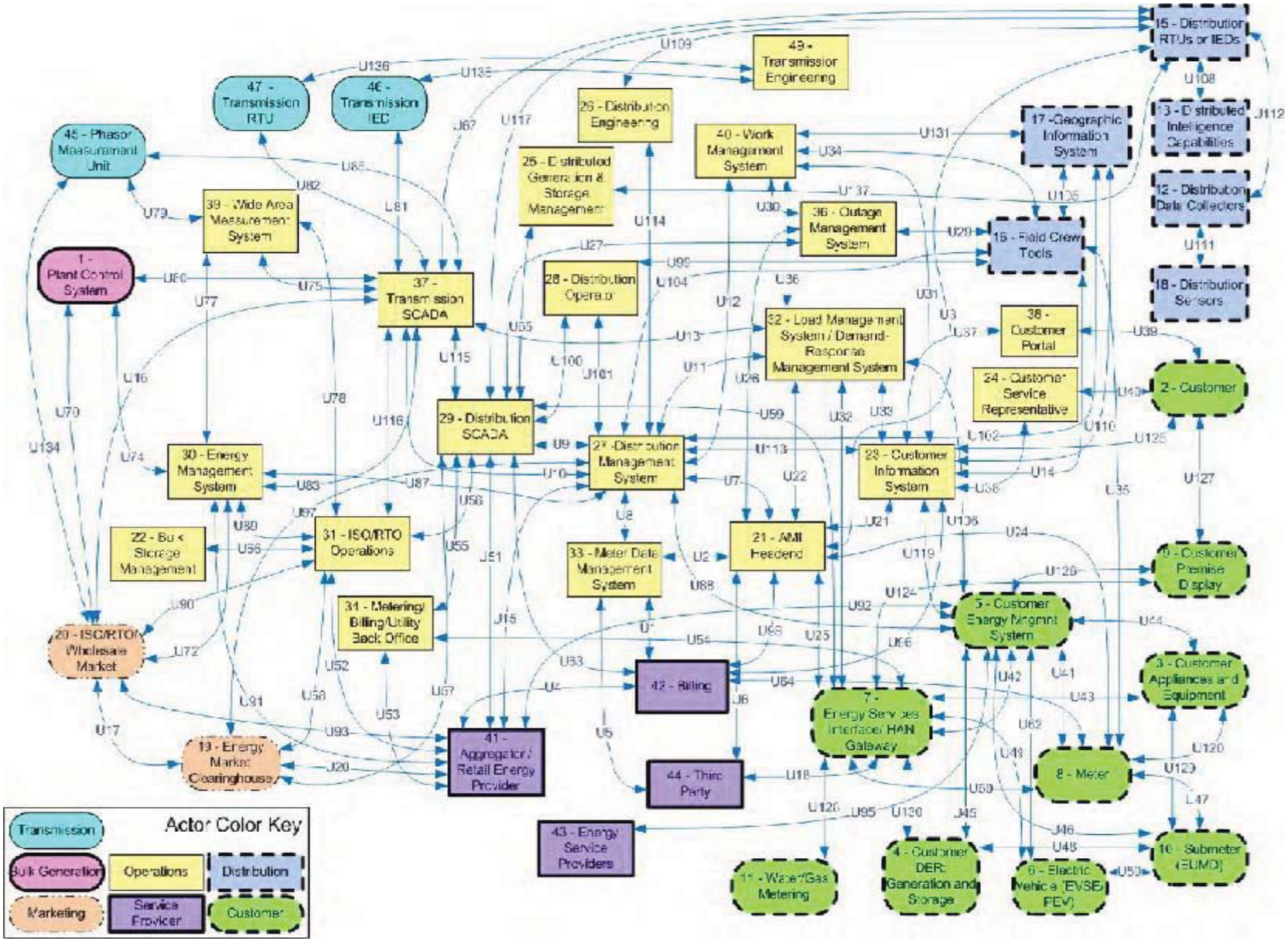
Reference: NIST IR 7628



Application Interfaces

- Applications interact within their own domain
- Applications interact across domains.
- Integration can be either
  - Point to Point or SOA Style

Reference: NIST IR 7628



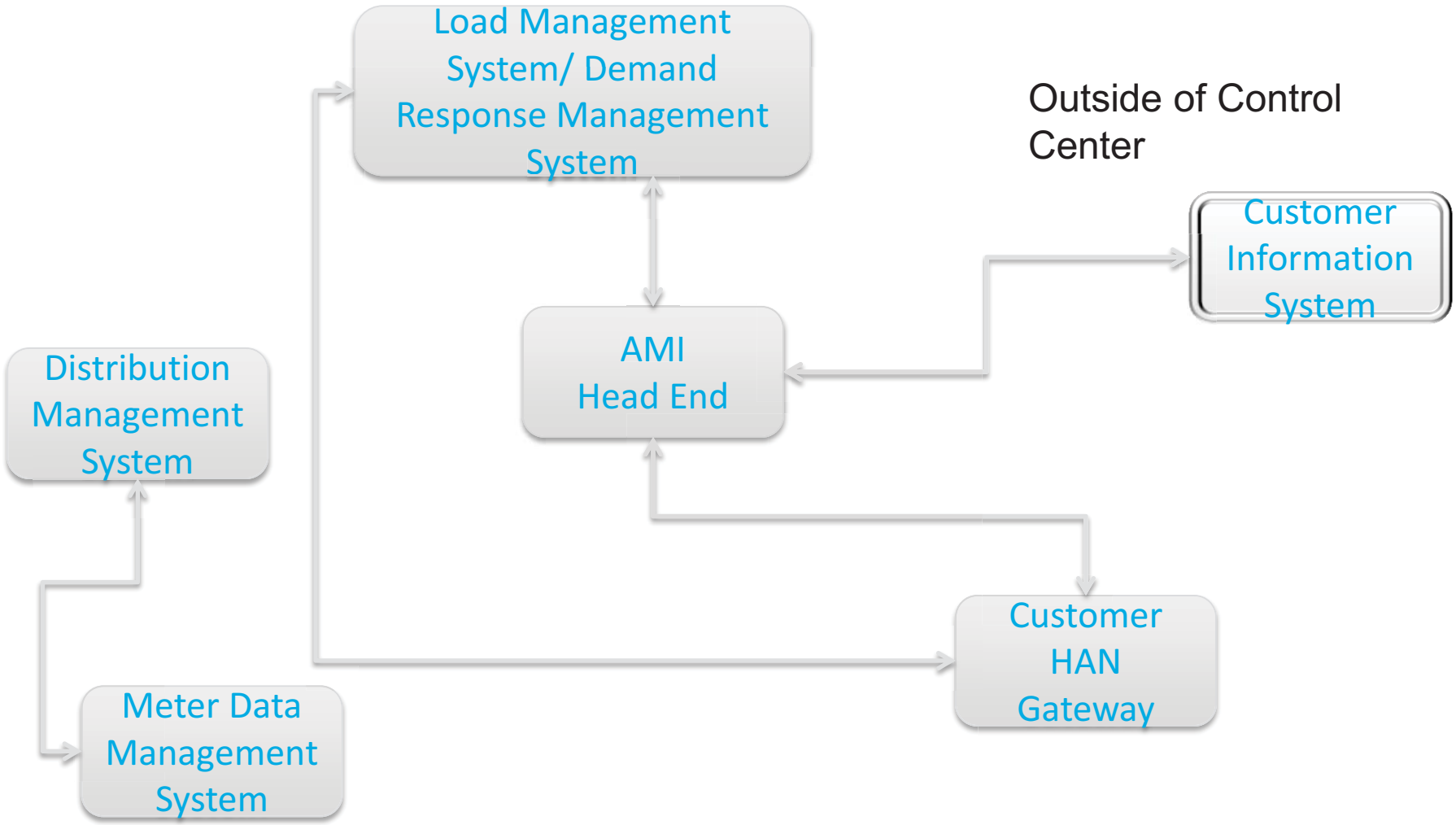
Smart Grid is System of Systems

- Individual software address specific need.
  - Demand Response, Energy Management system, Distribution Management System, Load forecast, AMI, etc.
- Applications fall within certain domain based on their functionality.
- Individual software are integrated to provide a solution.
  - OMS – AMI to Provide enhanced outage management System.
  - DRMS requires an AMI System to begin with.
- System of Systems – Individual sub system or system within a domain are integrated within and outside of the domain.

System Architecture for Integrating applications from various Smart Grid Domains

- Point to Point Integration
  - API Calls, Web Services and other native application support.
- Service Oriented Architecture
  - Message Based or Web Service based integration

Sub systems interactions



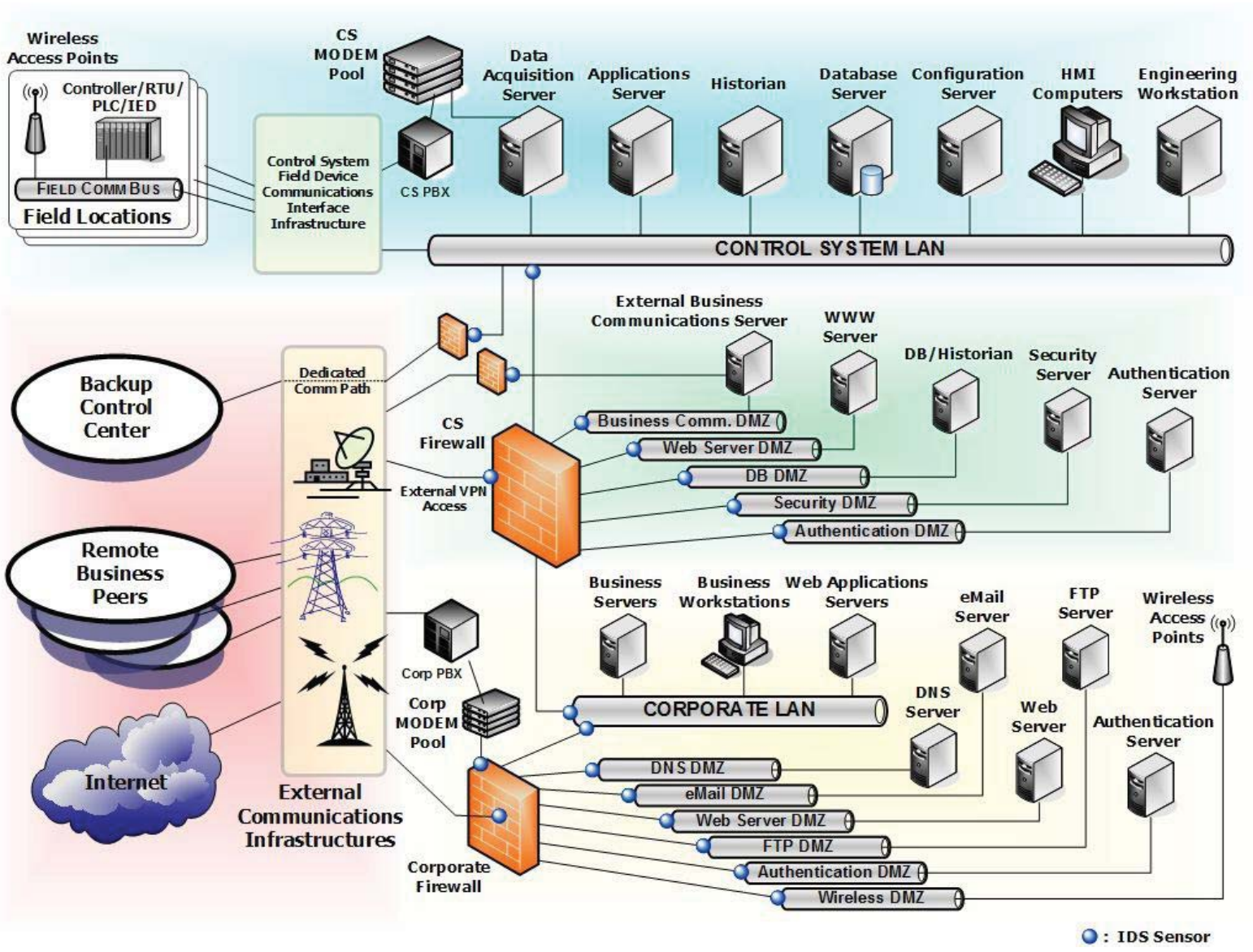
System Integration and Security Challenges

- System Integration is the weakest link (this was repeated several time in ICSJWG 2010)
- Only allow trusted systems should be allowed to access another system or service.
- Integrity & Non-repudiation of the data is very important.
  - Systems take actions based on message from another system.
- Confidentiality of certain critical data is important.
  - Passwords to access service is sent over the wire.
  - Example
    - CIS System accessing AMI Head end directly without even TLS in place is susceptible for “sniffing” and “man in the middle”.

Security challenges

- Is traditional Network Security good enough?
- System Architecture (Integration of applications in various domains) also have a role to play to ensure reliability and security.
  - E.g. When there are too many point to point integration, there are many firewall “holes” to poke.
- What about IDS/IPS? Even with Deep packet inspection and XML firewall capabilities?
- Intrusion Detection and Response cannot detect legitimate application usage but used by unauthorized users or systems.
  - *Reference: Richard Bejtlich, John Steven, Gunnar Peterson, "Directions in Incident Detection and Response," IEEE Security and Privacy, vol. 9, no. 1, pp. 91-92, Jan./Feb. 2011,* doi:10.1109/MSP.2011.6 *“Richard Bejtlich leads a conversation on how incident detection and response (IDR) teams' focus on detecting and preventing attacks has moved from* ***targeting OSs to unauthorized-access-application functionality and data****. He discusses why this makes IDR so much more difficult and what these new targets mean for IDR. “*
- Doesn’t mean that network security is not good / not needed. That is the start but not end all be all solution.

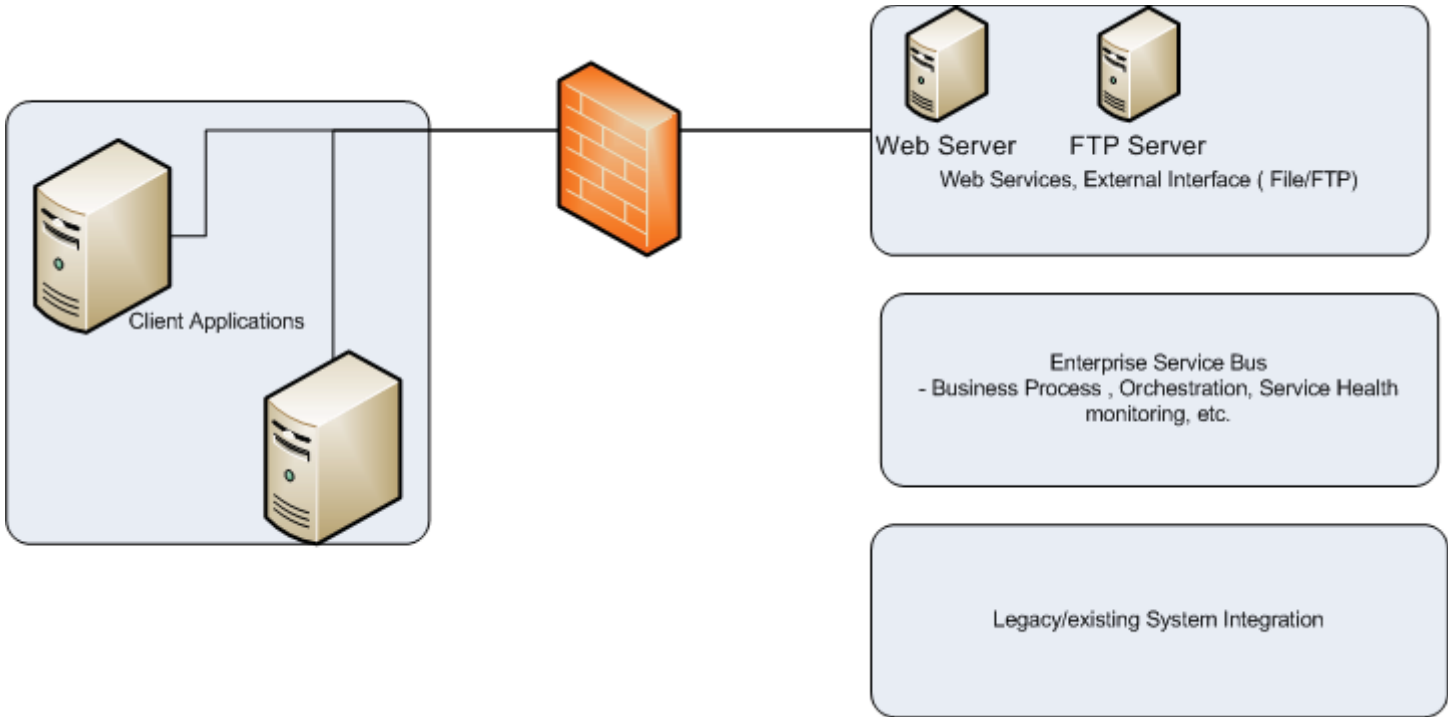
Reference: NIST SP 800-82



Software System Security challenges

- System Security starts with individual software, Integration layer capable of addressing security risks.
  - Authentication, Authorization, Confidentiality, Integrity & Non-repudiation.
- When Software is integrated and accessed as API/Service from distributed servers, interoperability is a key challenge.
- Understanding the threat /risk is very important.
  - This requires understanding how applications interact with one another.
  - What is the risk to reliability or safety when systems are compromised.
  - Would firewall, IDS/IPS and any other controls at perimeter good enough?
  - What is one system (e.g. CIS) that is in corporate domain is compromised? (e.g. password to access AMI services is stored in file which is not protected or encrypted).
    - This can lead to any intruder gaining access to services in another network domain.

Lets look at basic integration architecture



Threat modeling

- This is not a complete Threat analysis exercise.
- Let's start with Physical boundary. (AMI – CIS, DRMS – AMI, etc.)
- How hostile is the client applications environment compared to control center?
- Some of the Risks
- For Web Services , is HTTP Basic auth good enough? Or WS-Security with SOAP header is needed?
- Ensuring Integrity of the data.
  - Should the TLS /HTTPS good enough? Along with firewall and IDS/IPS?
  - What about TLS with Mutual authentication (HTTPS Mutual authentication). Isn't that good enough?
  - Message level integrity check to ensure there is no "man in the middle" changes.

Security Design Principles

- Externalize security from applications, components, services, etc.
  - Keeping it external will eliminate changes /testing when security functionality is added or modified.
- Software, System and Services shall support Interoperability.
- Integration with existing architecture , technology or solutions. (e.g. Integration with LDAP Server).
- Deny by default. Provide least privilege and add as necessary.
- Provide flexibility for administrator to configure, manage and enforce security.
- Policy Based security management and enforcement.
- Eye on Compliance and Industry Standards.
- Map such standards to technical requirements.

System Security Architecture Building blocks

- Centralized LDAP Server for authentication
- Authentication as Service
- Authorization as Service
- Provisioning for user life cycle management
- Interoperability Standards
- Confidentiality , Integrity and Non-repudiation at every layer
    - Physical or logical boundaries
    - Encryption should be aligned with access control
- Key Management should be integrated with overall System.
- Security as Service approach.

Benefits of System Security Architecture

- Automation removes errors and increases compliance.
- Security as a Service provides consistent way to integrate across products and services.
- Policy based ,externalized security provides way to modify/change policy without affecting operations.
- Ability to enhance security without having to change the products.

How to define and apply System Security Architecture

- Requirements are everywhere…
  - NERC CIP, NIST IR 7628, DHS Procurement language for control systems
  - None of these standards point out to a specific technology or solution approach. (not meant to)
- Create a consolidated technical requirements from
  - Various security standards
  - Internal security policy
- Define System Security Architecture vision and roadmap
  - Understand existing security technologies and how it is used/integrated. Pros and cons. Vendor support and vendor roadmap.
  - Create a “To be” architecture , vision that addresses your security risk and compliance.
  - Create a roadmap with plans for integration with existing and future products .

How to define and apply System Security Architecture contd..

- Integration with RFI/procurement process
  - Be as specific In RFI /RFP process
  - Translate your technical requirements and architecture into set of Questions.
  - Ask specific questions that reflect your policy, architecture and vision.
    - E.g. If you like to have centralized authentication against a specific LDAP Server, ask if the products support that LDAP Server.
    - If you internal policy (translated to technical requirements) requires products that expose Web Services to support WS-Security or HTTP Basic authentication, be specific on both near term and long term plans.
- Integrated Process
  - As industry/regulatory standards changes, update internal technical requirements
  - Update/review System Security Architecture. (Requirements, Vendor support and roadmap, internal policy)
  - Update RFI/RFP process as requirements and architecture changes.

Questions?

Sitaraman Lakshminarayanan

Security Architect – Systems Security

GE Energy

S.Lakshminarayanan@ge.com