

Workshop #2

***connected vehicle* Core System Architecture/Requirements**

September 20-22, 2011

San Jose, CA

Day 2

Systems Engineering Team

Wednesday

»» Welcome to Day 2

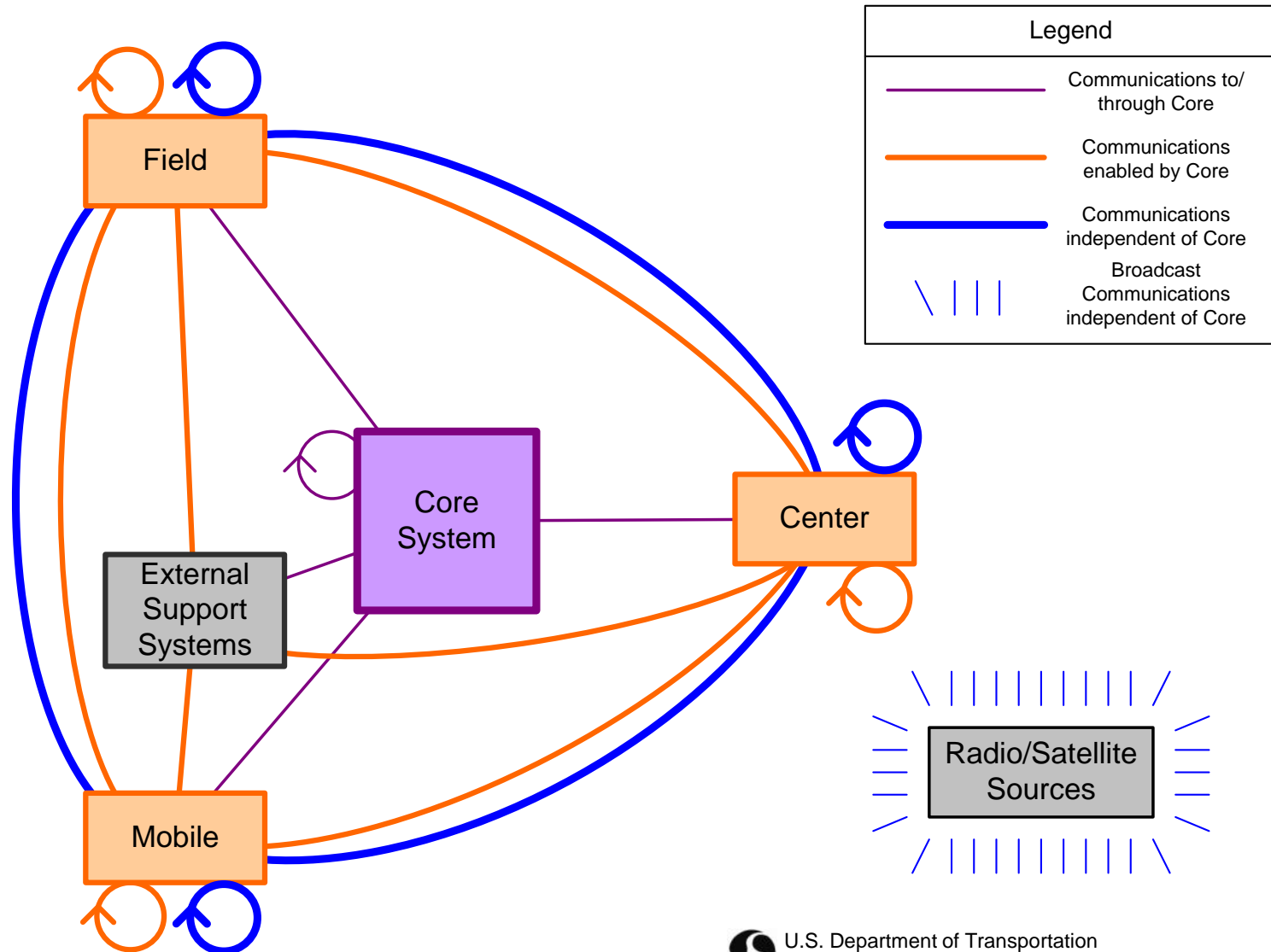
Agenda – Wednesday 9/21

9:00	Welcome & Recap
9:30	Architecture, Functional Views Discussion
10:15	Break
10:30	Architecture, Functional Views Discussion
12:00	Lunch
1:15	Architecture, Functional Views Discussion
2:30	Break
2:45	Architecture, Connectivity Views Discussion
4:30	Adjourn for the day

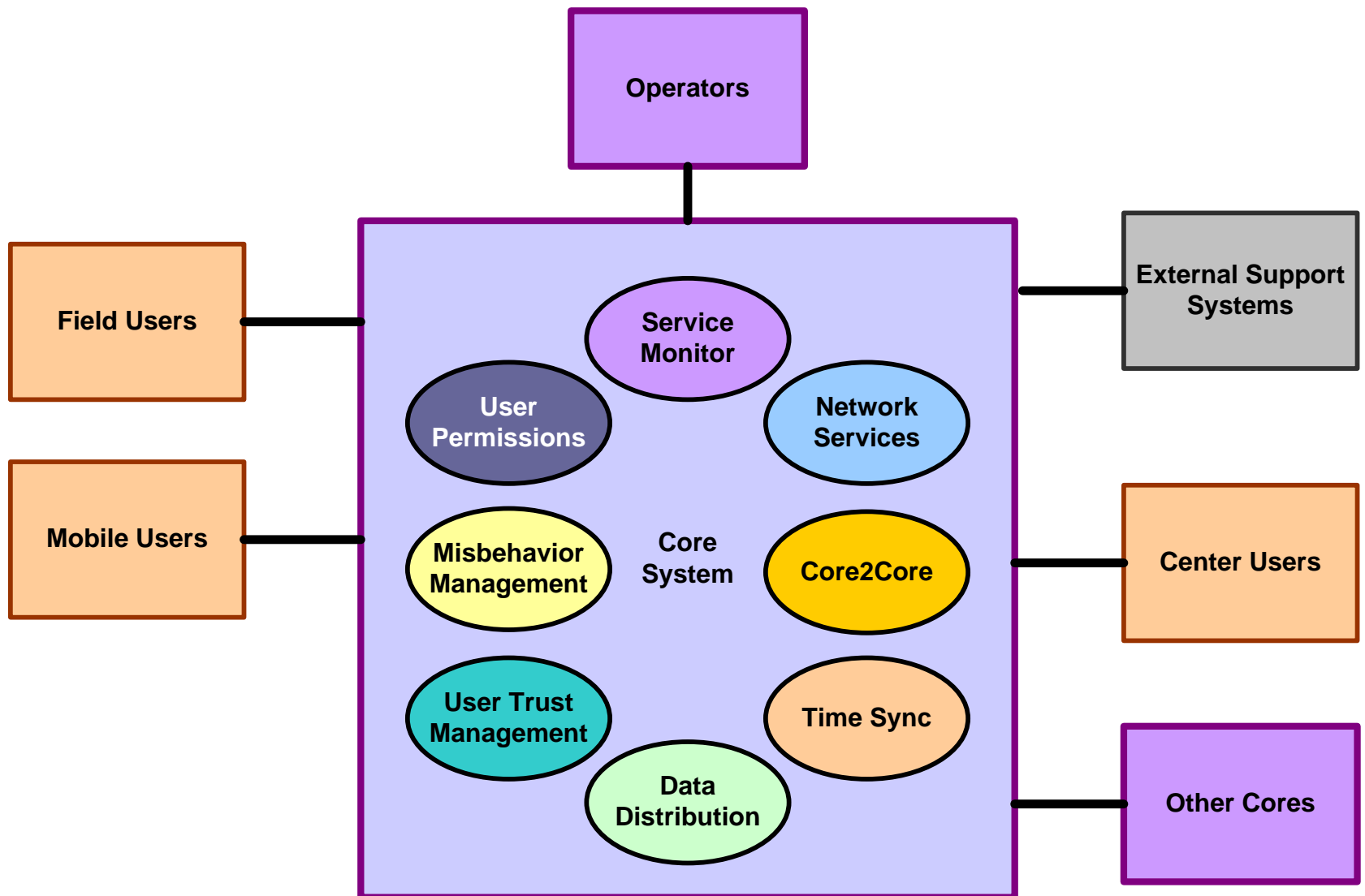
Core System provides services that...

- Enable data transfers between system users
 - Mobile
 - Field
 - Center
- Are in a secure, trusted environment
 - Enabling trust between parties that have no direct relationship
 - Enabling secure data exchange between parties that have no direct relationship
 - Enabling the exchange of data between parties that have data and parties that want data

Core System in the context of the *connected vehicle* environment



Core System's 8 Subsystems



Core System Architecture: Functional Views

- » Views that will describe the logical interactions

Functional

- Focuses on the behavior, structure, and interaction of the functions performed by the system
- Shows functions for each subsystem
- Traceable to *functional* requirements
- Color coding:
 - Subsystems each represented by a different color
 - Information Objects are the same color as the source Function object



Functional

- For each view:
 - Description
 - Consideration/Concerns
 - Entities and their relationships (diagram)
 - Alternatives explored
 - Other related views



Functional

Check User
Permission

Functional Objects

Parse Data

Functional Objects that
represent optional functions

Distribute
CRL

Functional Objects that
store encrypted data

Data
Acceptance
Catalog

Data stores

Logical relationships between
Functional Objects

Data flow between a Functional
Object and an external Object

User Identification

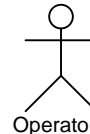
Information Object

Data

An Information Object whose sender
expects an acknowledgement

Misbehavior
Report

An Information Object that is secure



Operator

An external actor

Functional Views Defined

- Top Level
- Data Distribution
- System Configuration
- User Configuration
- System Monitor and Control
- Credentials Distribution
- Misbehavior Management
- Core Decryption
- Networking
- Core Backup



Functional View 4.2.1 – Top Level

- Description:
 - Objects map to the subsystems in the Concept of Operations,
 - Provide the basis for all subsequent functional views
 - Satisfy functional requirements
 - Illustrate interface requirements
 - Specifies which subsystems provide external interfaces

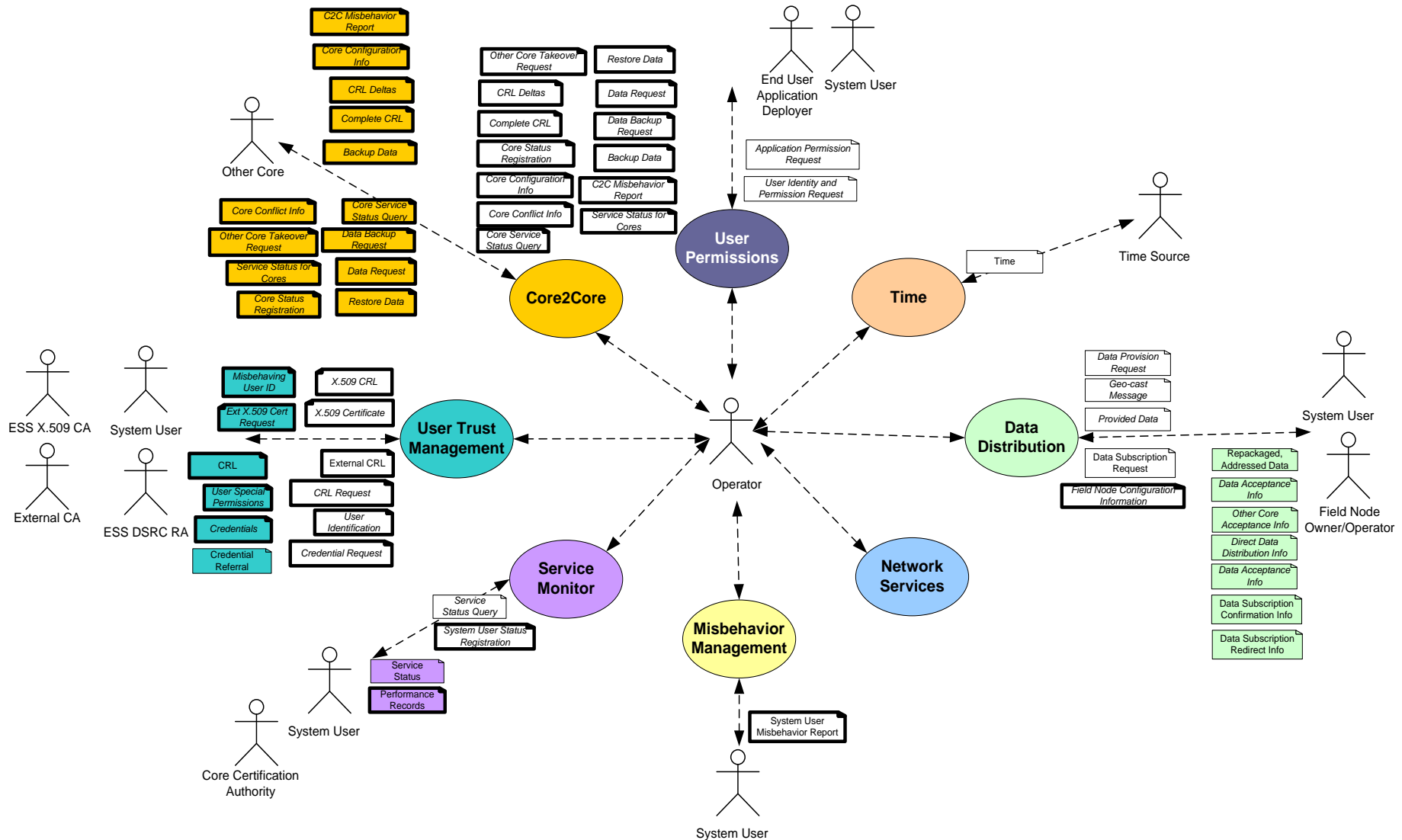
Functional View 4.2.1 – Top Level

■ Considerations/Concerns Addressed:

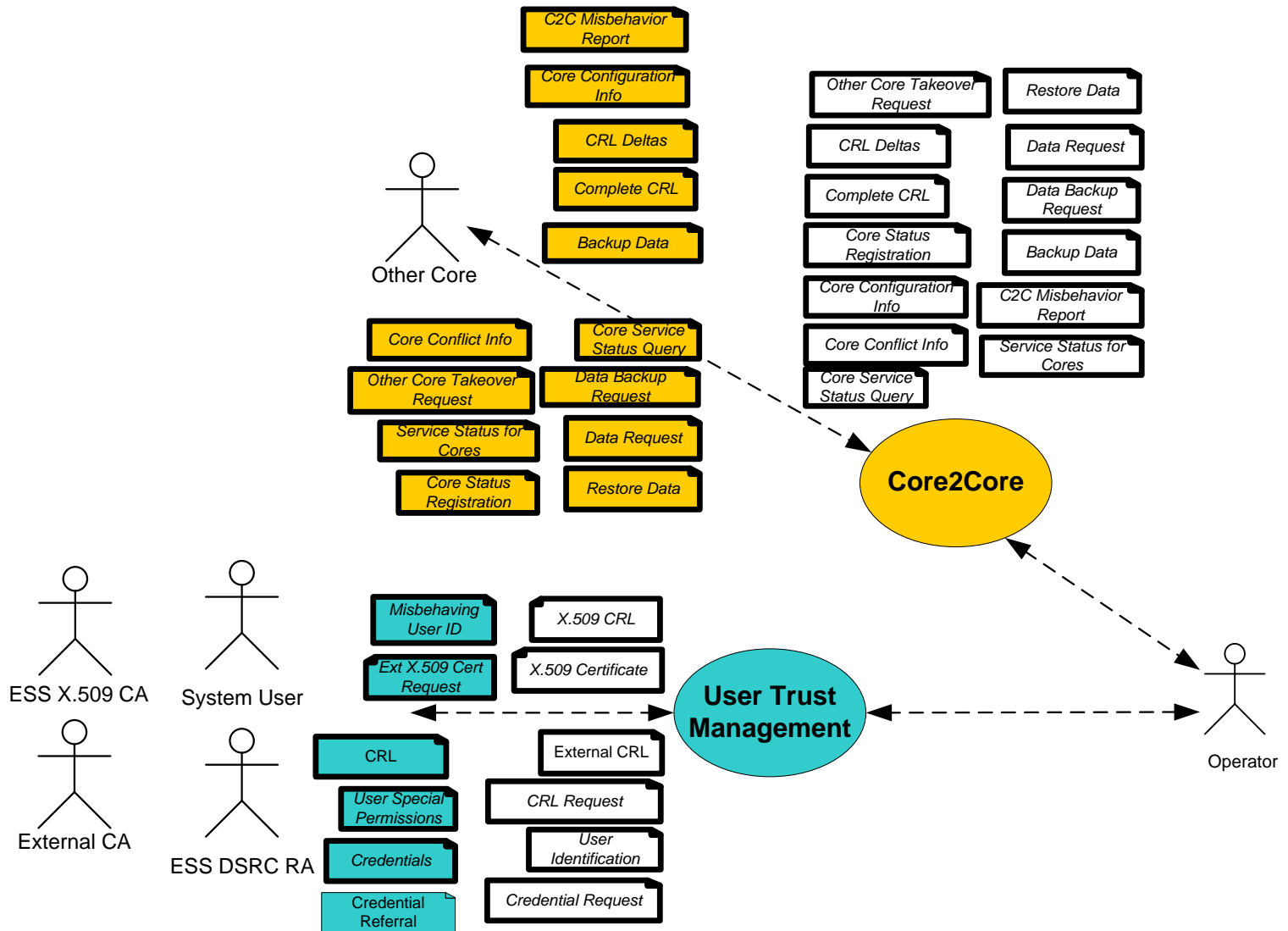
Functionality	<p>How does the Core System monitor the services it provides?</p> <p>How does the Core System support the coordination of resources between different Cores?</p> <p>How does the Core System function internally?</p> <p>How does the Core System transition between operational modes?</p>
Security	<p>What functional elements are involved in the distribution and revocation of digital certificates, and what roles do those entities have?</p> <p>What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have?</p>
Appropriateness	<p>Does the Core System meet all of the needs defined in the ConOps?</p> <p>Does the Core System meet all of the functional requirements defined in the SyRS?</p>



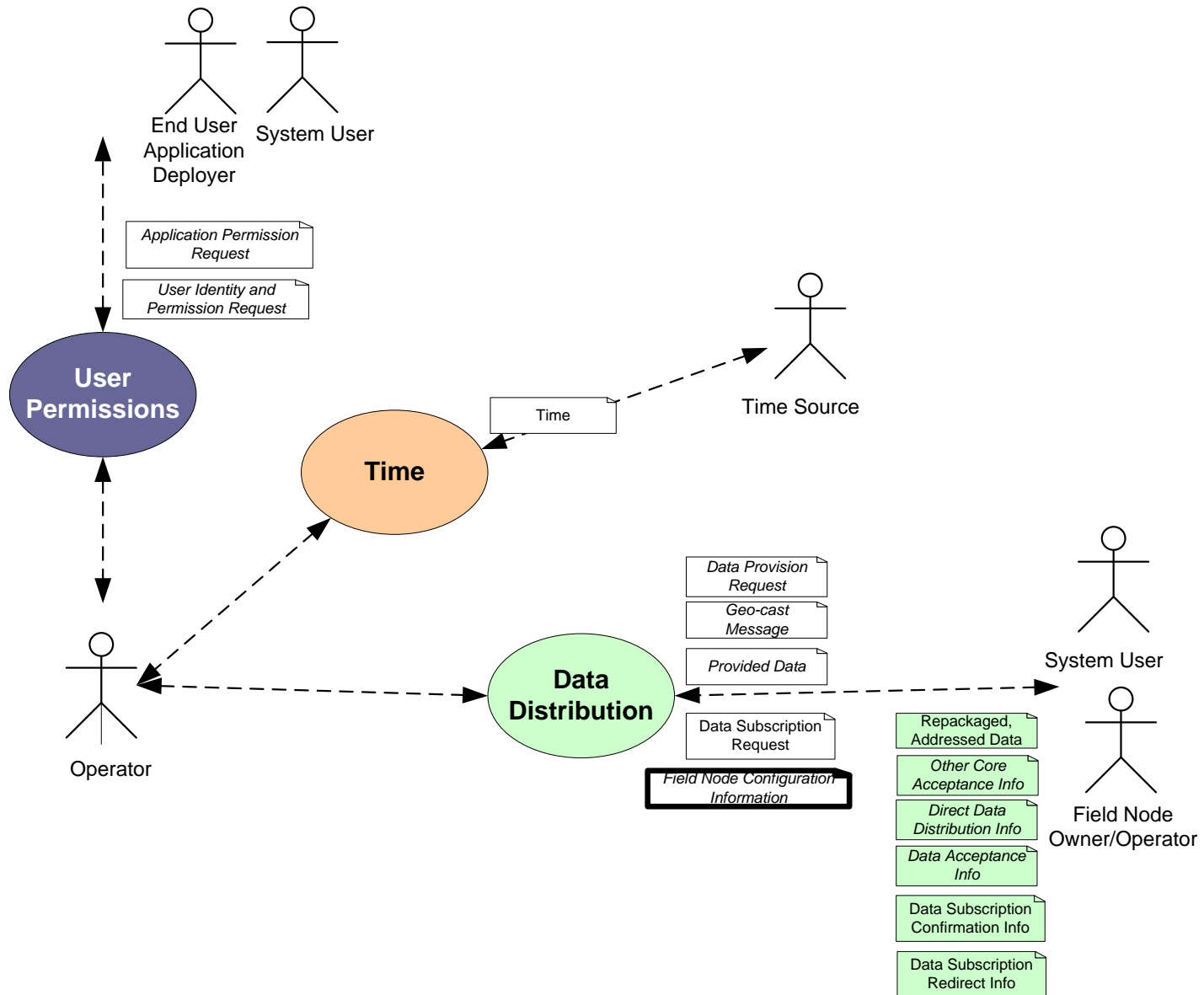
Functional View 4.2.1 – Top Level



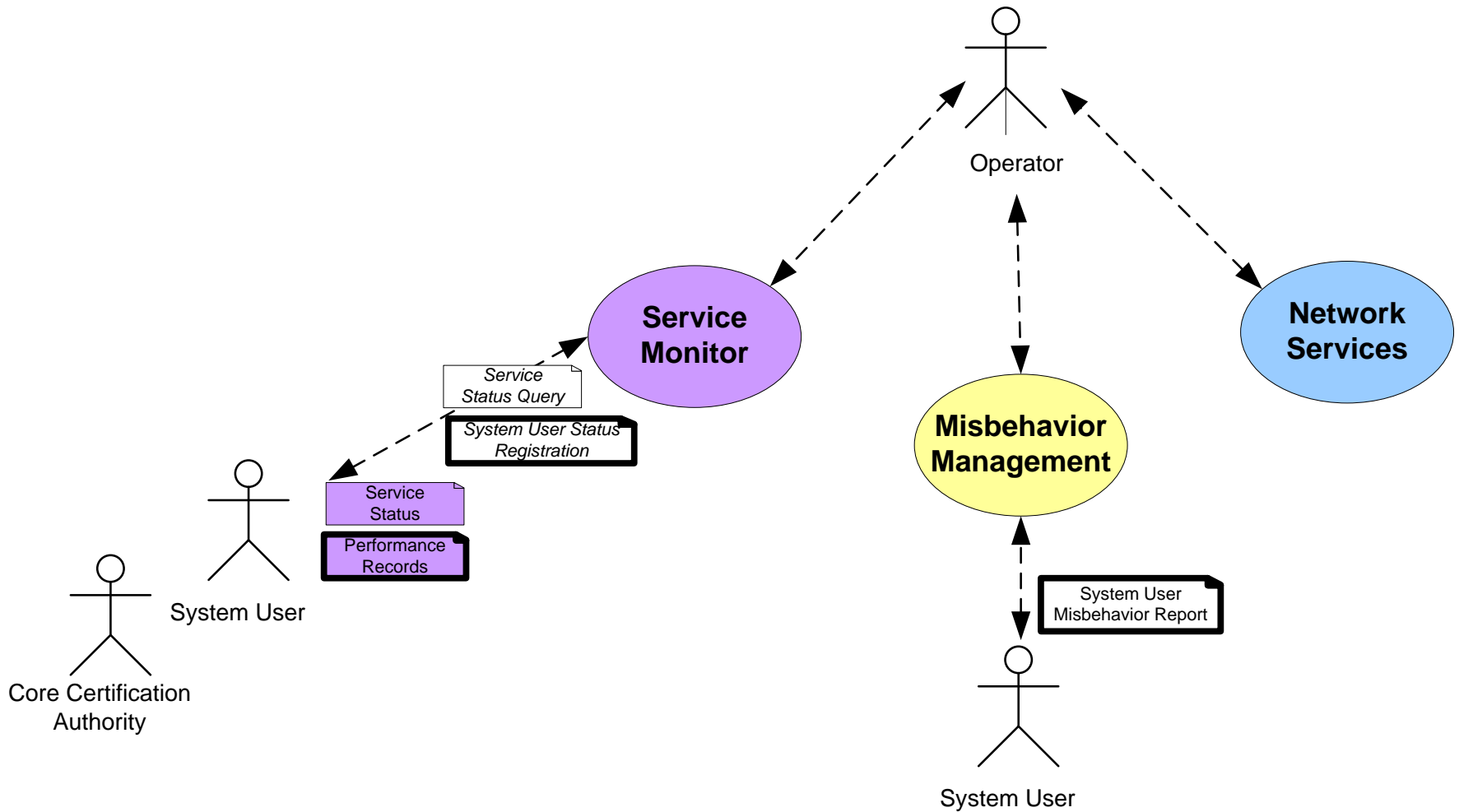
Functional View 4.2.1 – Top Level



Functional View 4.2.1 – Top Level



Functional View 4.2.1 – Top Level



Functional View 4.2.1 – Top Level

Related Views:

- Functional Views
 - Data Distribution
 - System Configuration
 - User Configuration
 - System Monitor and Control
 - Credentials Distribution
 - Misbehavior Management
 - Core Decryption
 - Networking
 - Core Backup
- Connectivity Views
 - High Level
 - Core System Functional Allocation
- Information Views
 - Top Level External Objects

Functional View 4.2.2 – Data Distribution

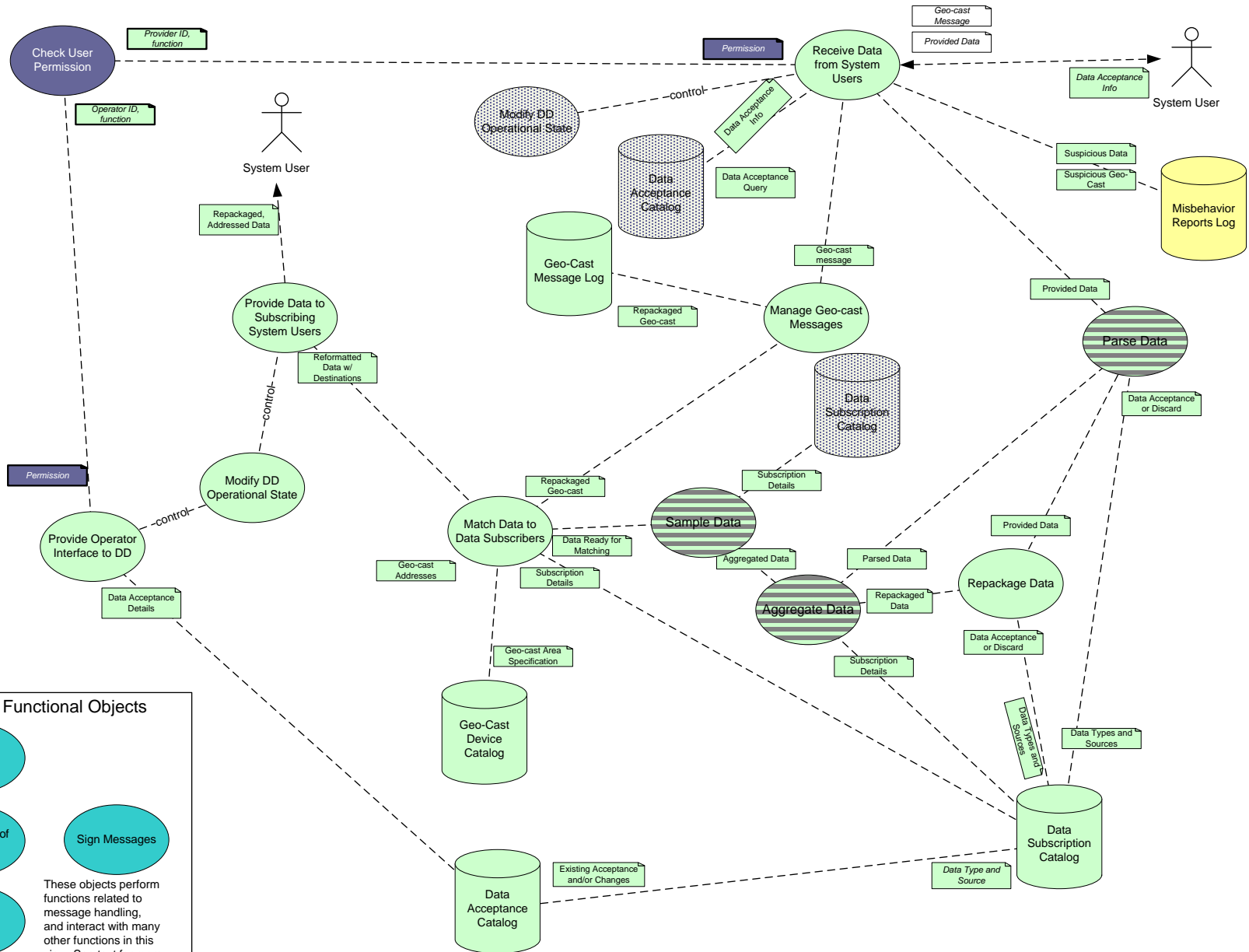
- Description:
 - System Users provide data, other System Users subscribe to data;
 - The Core matches those providers and consumers without requiring them to enter into a relationship with the other
 - Includes several optional functions: data aggregation, data parsing and data sampling

Functional View 4.2.2 – Data Distribution

■ Considerations/Concerns Addressed:

Interfaces	<p>How difficult is it to develop applications that use Core System interfaces?</p> <p>How flexible are Core System data distribution interfaces?</p> <p>How does the Core System enable control of the services it provides?</p>
Functionality	<p>How does the Core System monitor the services it provides?</p> <p>How does the Core System function internally?</p> <p>How do the Core System's components work together?</p> <p>How does the Core System transition between operational modes?</p>
Security	<p>What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have?</p> <p>How does the Core System maintain the integrity of information provided to it by System Users?</p>
Appropriateness	<p>Does the Core System meet all of the needs defined in the ConOps?</p> <p>Does the Core System meet all of the functional requirements defined in the SyRS?</p>
Evolvability	<p>How easily can the Core's functionality be expanded to cover new needs if they arise?</p> <p>Does the functionality of the Core scale to support foreseeable demands from System Users?</p>

Functional View 4.2.2 – Data Distribution



Functional View 4.2.2 – Data Distribution

- Alternatives Considered:
 - An alternative that omitted the parsing and repackaging function was reviewed at the June workshop
 - Deployers of a Core System might weigh the communications requirements to send large blocks of raw data against the computing requirements to process the queries for data.
 - Result was to make the parsing, aggregating, and sampling functions optional



Functional View 4.2.2 – Data Distribution

Related Views:

- Enterprise Views
 - Operations
 - Governance
- Functional Views
 - Top Level
 - System Configuration
 - User Configuration
 - Core Backup
- Connectivity Views
 - Core System
Functional Allocation
- Information Views
 - Top Level External
Objects
 - Top Level Internal
Objects

Functional View 4.2.3 – System Configuration

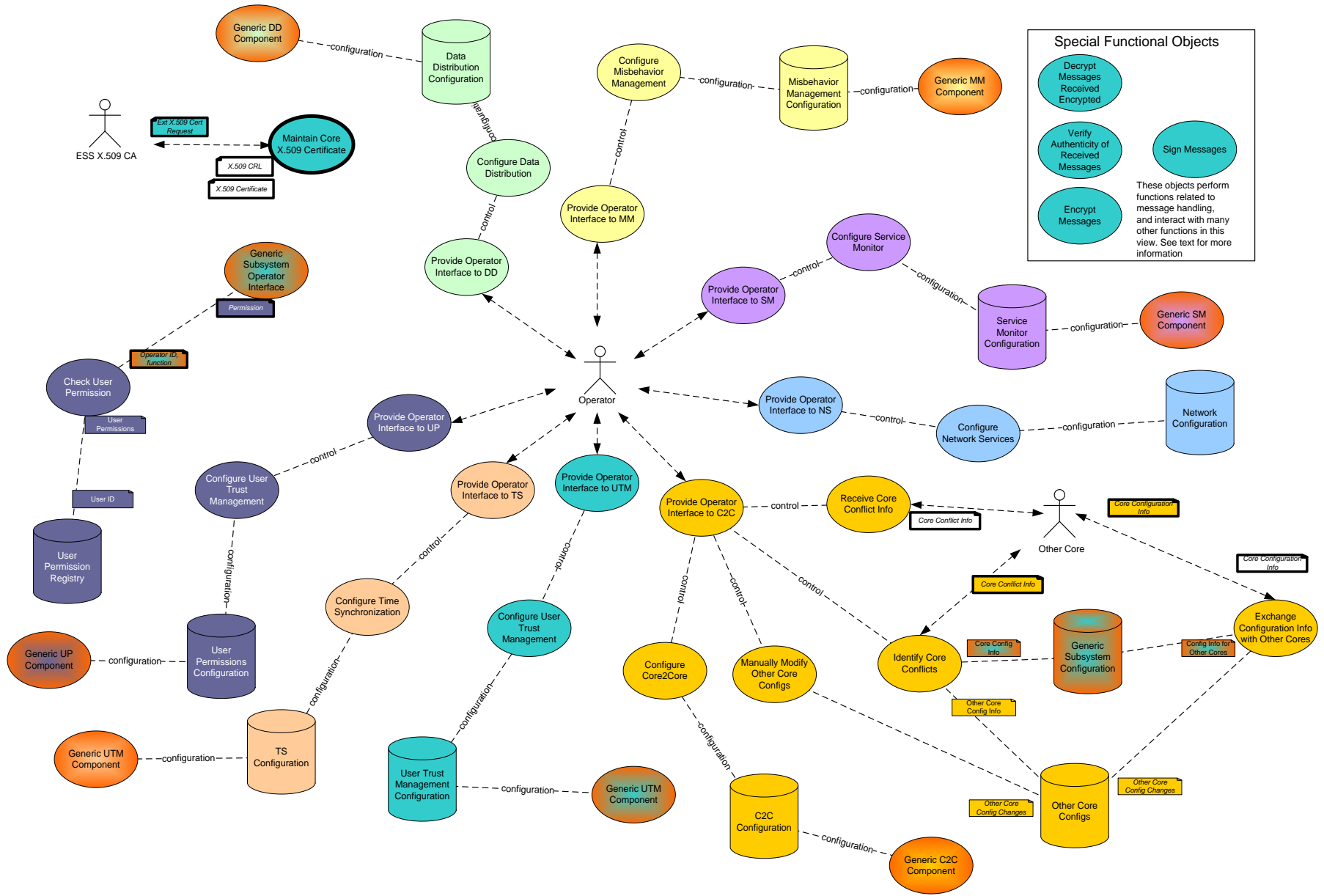
- Description:
 - Addresses the configuration of all Core subsystems, both for installation and changes to configuration
 - Maintain an understanding of the configuration of other Cores
 - Enables a variety of Core System configurations

Functional View 4.2.3 – System Configuration

■ Considerations/Concerns Addressed:

Functionality	<p>How does the Core System support the coordination of resources between different Cores?</p> <p>How does the Core System function internally?</p> <p>How do the Core System's components work together?</p> <p>How does the Core System transition between operational modes?</p>
Appropriateness	<p>Does the Core System meet all of the needs defined in the ConOps?</p> <p>Does the Core System meet all of the functional requirements defined in the SyRS?</p>
Evolvability	<p>How easily can the Core's functionality be expanded to cover new needs if they arise?</p> <p>Does the functionality of the Core scale to support foreseeable demands from System Users?</p>

Functional View 4.2.3 – System Configuration



Functional View 4.2.3 – System Configuration

Related Views:

- Enterprise Views
 - Operations
 - Configuration and Maintenance
 - Governance
- Functional Views
 - Top Level
 - User Configuration
 - System Monitor & Control
 - Core Backup
- Connectivity Views
 - Core System Functional Allocation
- Information Views
 - Top Level External Objects
 - Top Level Internal Objects

Functional View 4.2.4 – User Configuration

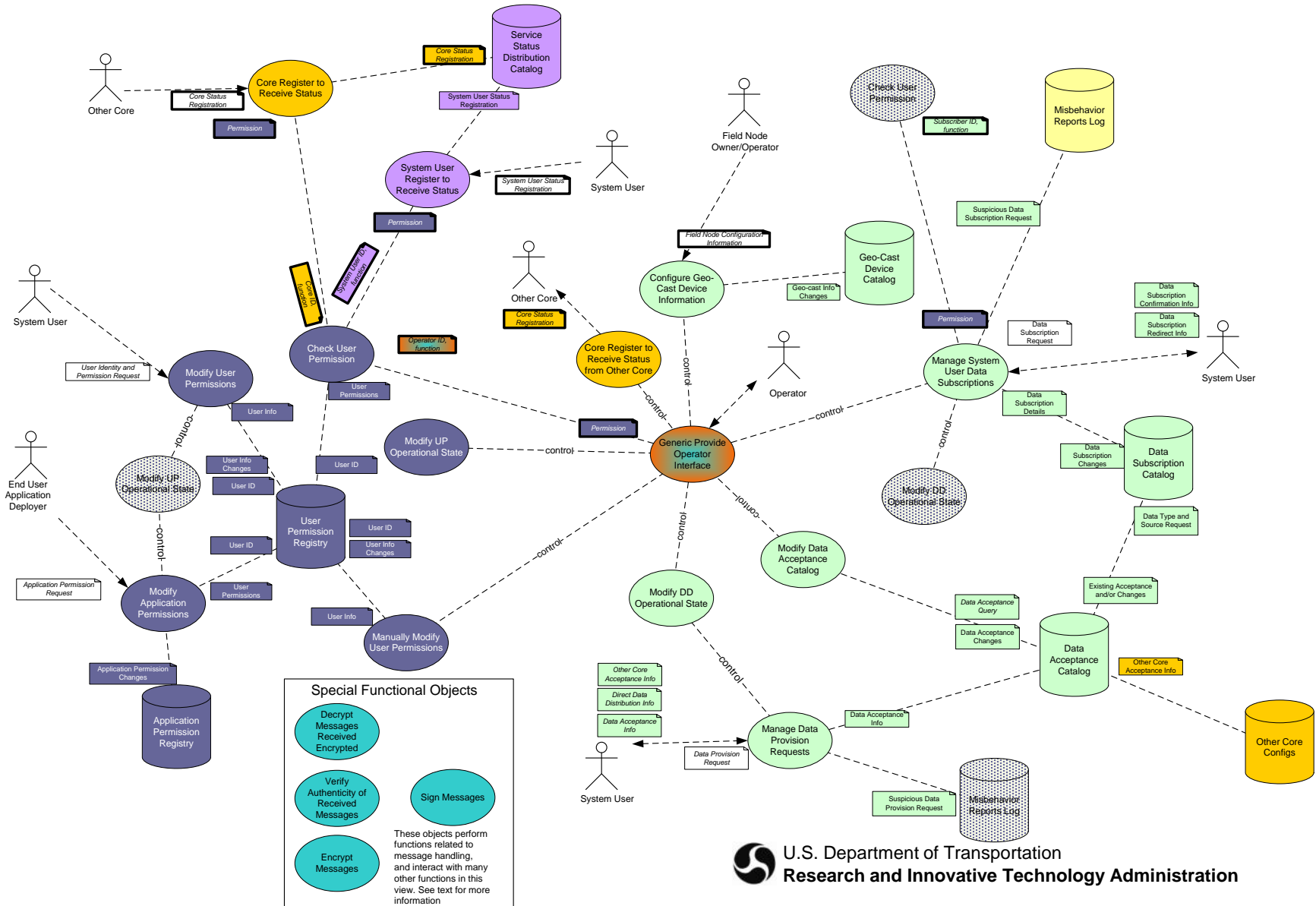
- Description:
 - Addresses the configuration of Core System user accounts and their data subscriptions
 - Manages Operator, System User, and Core requests for periodic service status updates
 - Creation and modification of System User data provision requests and subscriptions
 - Including actions by Operators, System Users and other Cores

Functional View 4.2.4 – User Configuration

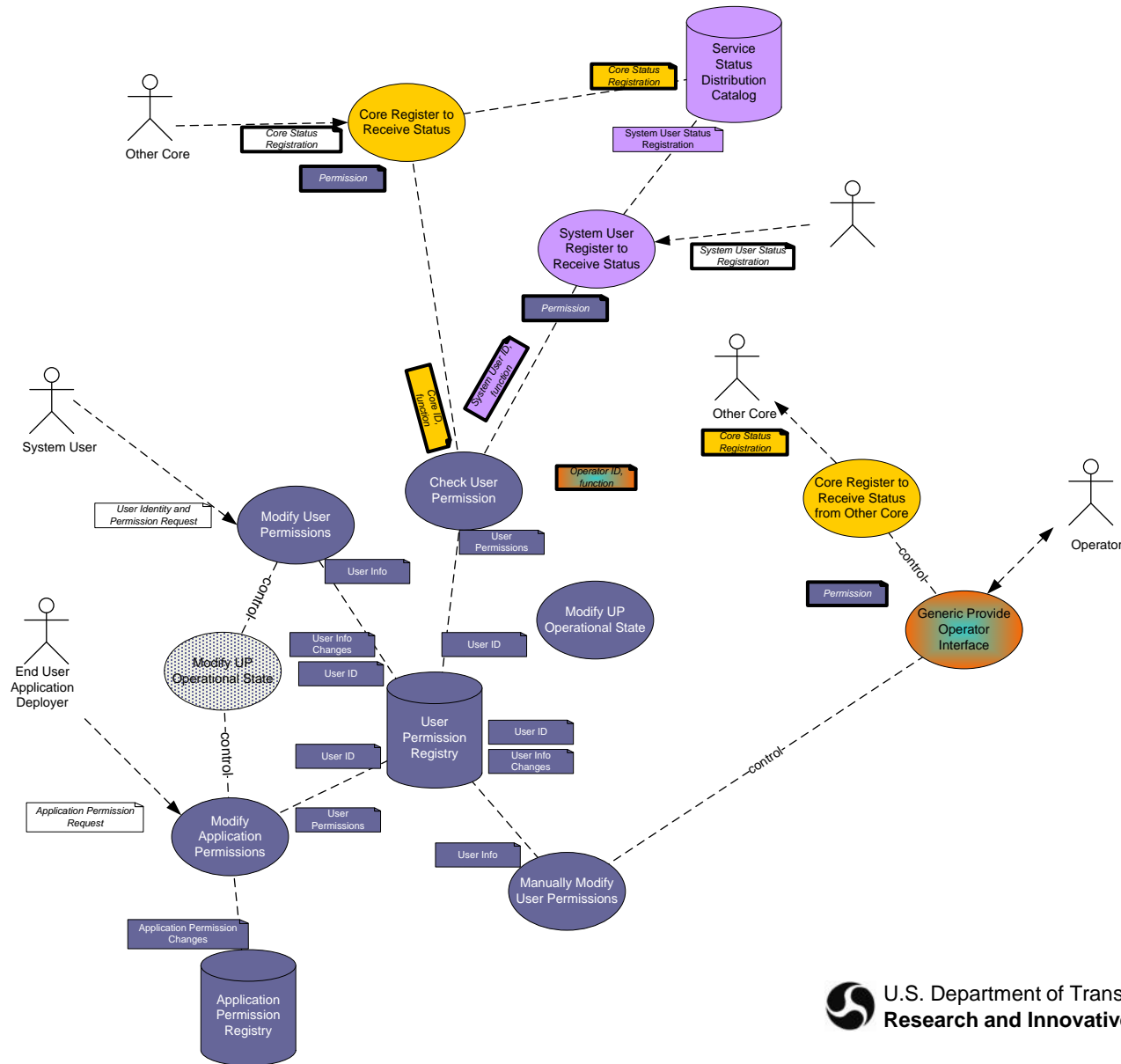
■ Considerations/Concerns Addressed:

Interfaces	<p>How flexible are Core System data distribution interfaces?</p> <p>How does the Core System enable control of the services it provides?</p>
Functionality	<p>How does the Core System monitor the services it provides?</p> <p>How does the Core System support the coordination of resources between different Cores?</p> <p>How does the Core System function internally?</p> <p>How do the Core System’s components work together?</p> <p>How does the Core System transition between operational modes?</p>
Security	<p>What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have?</p> <p>How does the Core System maintain the integrity of information provided to it by System Users?</p> <p>How does the Core System secure System Users’ personal information?</p>
Appropriateness	<p>Does the Core System meet all of the needs defined in the ConOps?</p> <p>Does the Core System meet all of the functional requirements defined in the SyRS?</p>
Evolvability	<p>How easily can the Core’s functionality be expanded to cover new needs if they arise?</p> <p>Does the functionality of the Core scale to support foreseeable demands from System Users?</p>

Functional View 4.2.4 – User Configuration



Functional View 4.2.4 – User Configuration

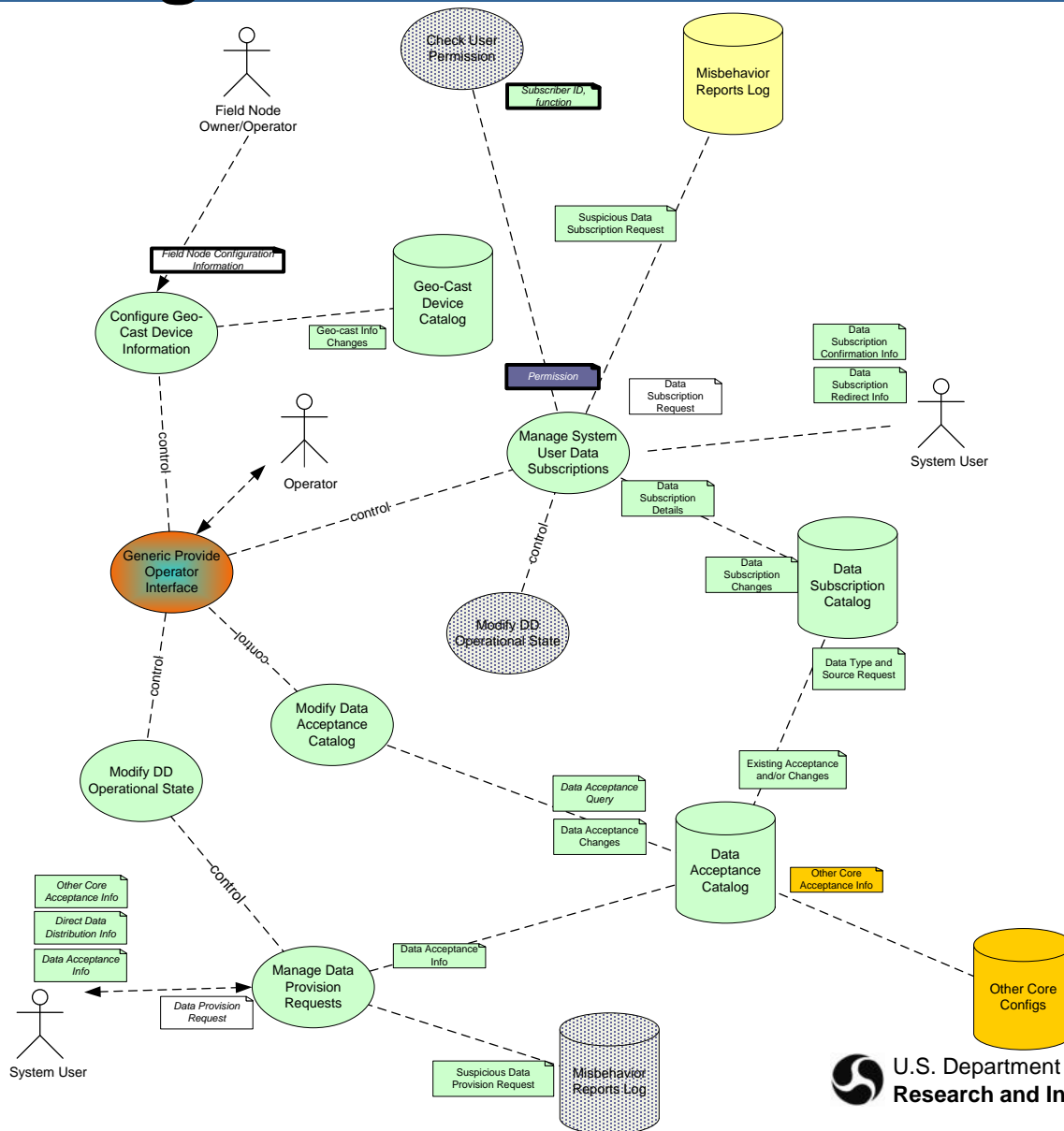


Special Functional Objects

- Decrypt Messages Received Encrypted
- Verify Authenticity of Received Messages
- Encrypt Messages
- Sign Messages

These objects perform functions related to message handling, and interact with many other functions in this view. See text for more information

Functional View 4.2.4 – User Configuration



Functional View 4.2.4 – User Configuration

Related Views:

- Enterprise Views
 - Operations
 - Configuration and Maintenance
 - Governance
- Functional Views
 - Top Level
 - System Configuration
 - System Monitor & Control
- Connectivity Views
 - Core System Functional Allocation
- Information Views
 - Top Level External Objects
 - Top Level Internal Objects

Functional View 4.2.5 – System Monitor & Control

- Description:
 - Day-to-day housekeeping functions that enable the Operator to manage the Core System's operations
 - Monitoring of subsystem anomalies and state changes
 - Operator interfaces
 - Monitor the environmental conditions the Core operates in
 - Provide status to operators, some system users
 - Supports maintenance actions

Functional View 4.2.5 – System Monitor & Control

■ Considerations/Concerns Addressed:

Functionality	<p>How does the Core System monitor the services it provides?</p> <p>How does the Core System function internally?</p> <p>How do the Core System's components work together?</p> <p>How does the Core System transition between operational modes?</p>
Appropriateness	<p>Does the Core System meet all of the needs defined in the ConOps?</p> <p>Does the Core System meet all of the functional requirements defined in the SyRS?</p>

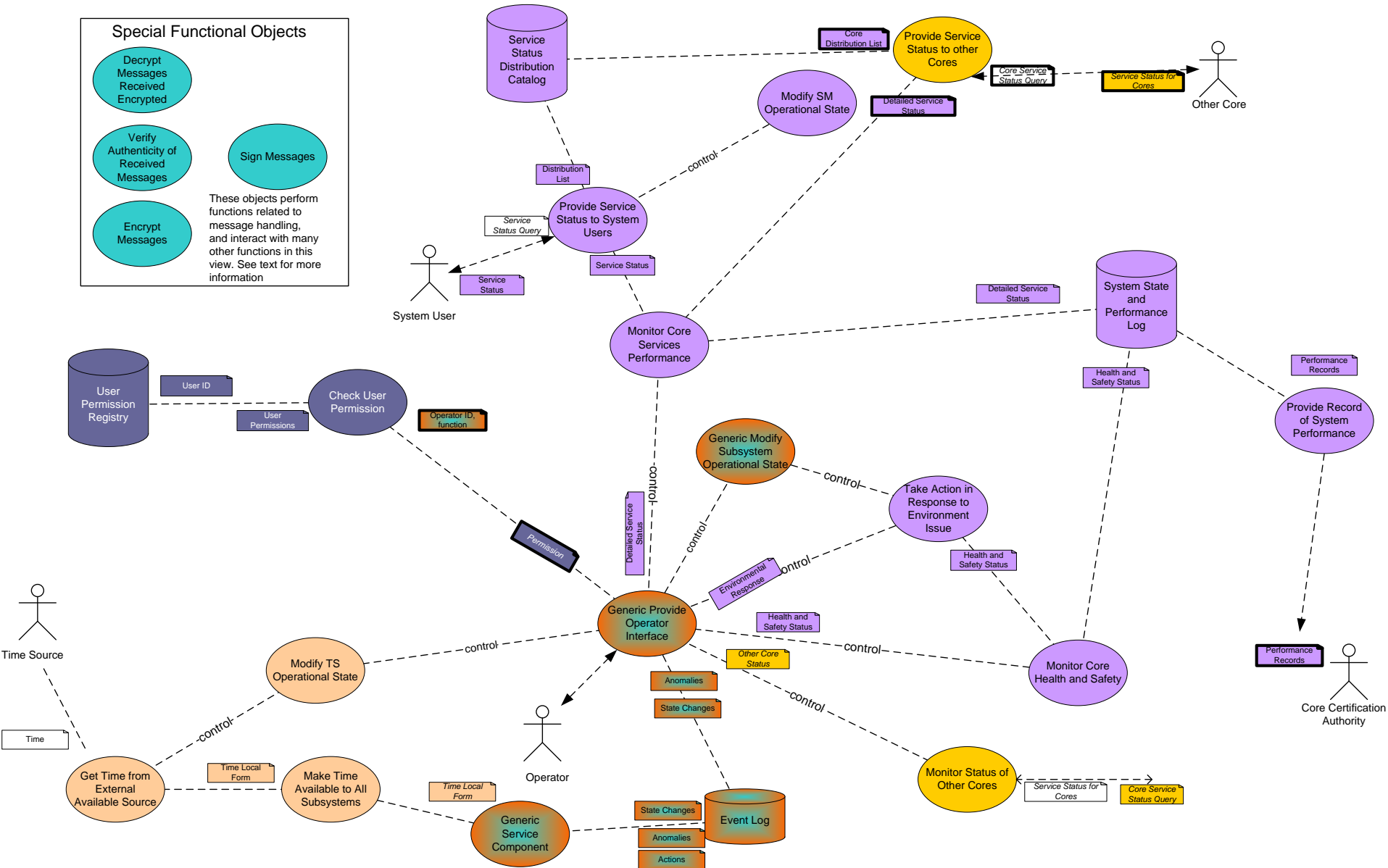


Functional View 4.2.5 – System Monitor & Control

Special Functional Objects

- Decrypt Messages Received Encrypted
- Verify Authenticity of Received Messages
- Sign Messages
- Encrypt Messages

These objects perform functions related to message handling, and interact with many other functions in this view. See text for more information



Functional View 4.2.5 – System Monitor & Control

Related Views:

- Enterprise Views
 - Operations
 - Configuration and Maintenance
 - Governance
- Functional Views
 - Top Level
 - System Configuration
 - User Configuration
 - Core Backup
- Connectivity Views
 - Core System Functional Allocation
- Information Views
 - Top Level External Objects
 - Top Level Internal Objects

Functional View 4.2.6 – Credentials Distribution

- Description:
 - Ensure Trust with Field and Center System Users and with Other Cores
 - Mobile User credentials handled outside Core
 - Management of credentials, including X.509 digital certificates, CRLs, and assignment and recognition of credential-related roles (i.e., registration versus certificate distribution)

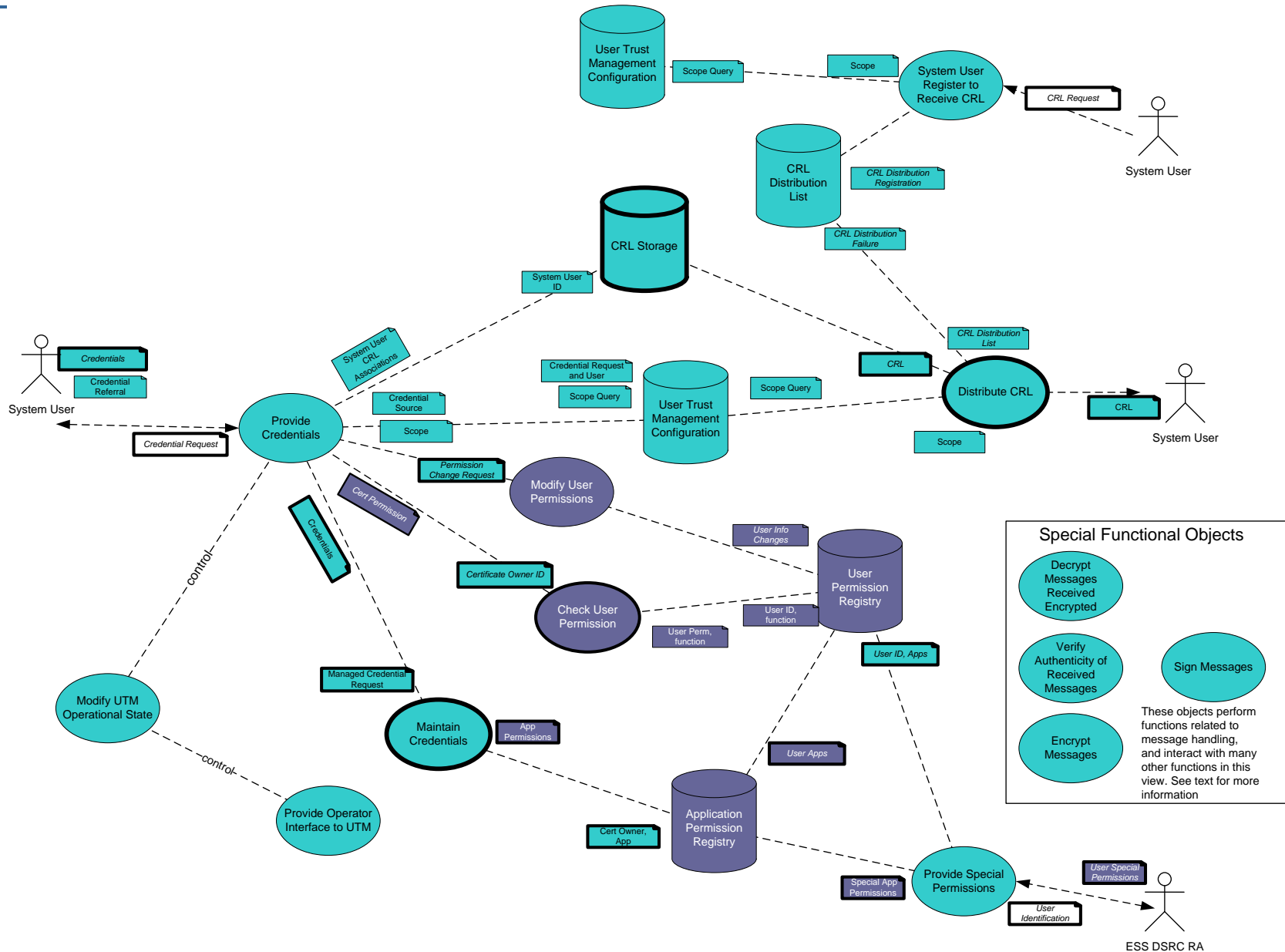
Functional View 4.2.6 – Credentials Distribution

■ Considerations/Concerns Addressed:

<p>Functionality</p>	<p>How does the Core System monitor the services it provides?</p> <p>How does the Core System support the coordination of resources between different Cores?</p> <p>How does the Core System function internally?</p> <p>How do the Core System’s components work together?</p> <p>How does the Core System transition between operational modes?</p>
<p>Security</p>	<p>What functional elements are involved in the distribution and revocation of digital certificates, and what roles do those entities have?</p> <p>How does the Core System maintain the integrity of information provided to it by System Users?</p> <p>How does the Core System maintain the privacy of communications between System Users?</p> <p>How does the Core System secure System Users’ personal information?</p>
<p>Appropriateness</p>	<p>Does the Core System meet all of the needs defined in the ConOps?</p> <p>Does the Core System meet all of the functional requirements defined in the SyRS?</p>
<p>Evolvability</p>	<p>Does the functionality of the Core scale to support foreseeable demands from System Users?</p>



Functional View 4.2.6 – Credentials Distribution



Functional View 4.2.6 – Credentials Distribution

- Alternatives Considered:
 - Discussed previously in Enterprise Views for Credential Distribution
 - Functional alternatives not selected included features:
 - Core as a CA,
 - Core as a CA, with pre-loaded Certs
 - Separate RA and CA Cores
 - External CA for anonymous DSRC Certs only
 - Multiple root CAs

Functional View 4.2.6 – Credentials Distribution

Related Views:

- Enterprise Views
 - Security Credentials
 - Governance
 - Business Model Facilitation
- Functional Views
 - Top Level
 - Misbehavior Management
- Connectivity Views
 - Core System Functional Allocation
- Information Views
 - Top Level External Objects
 - Top Level Internal Objects

Functional View 4.2.7 – Misbehavior Management

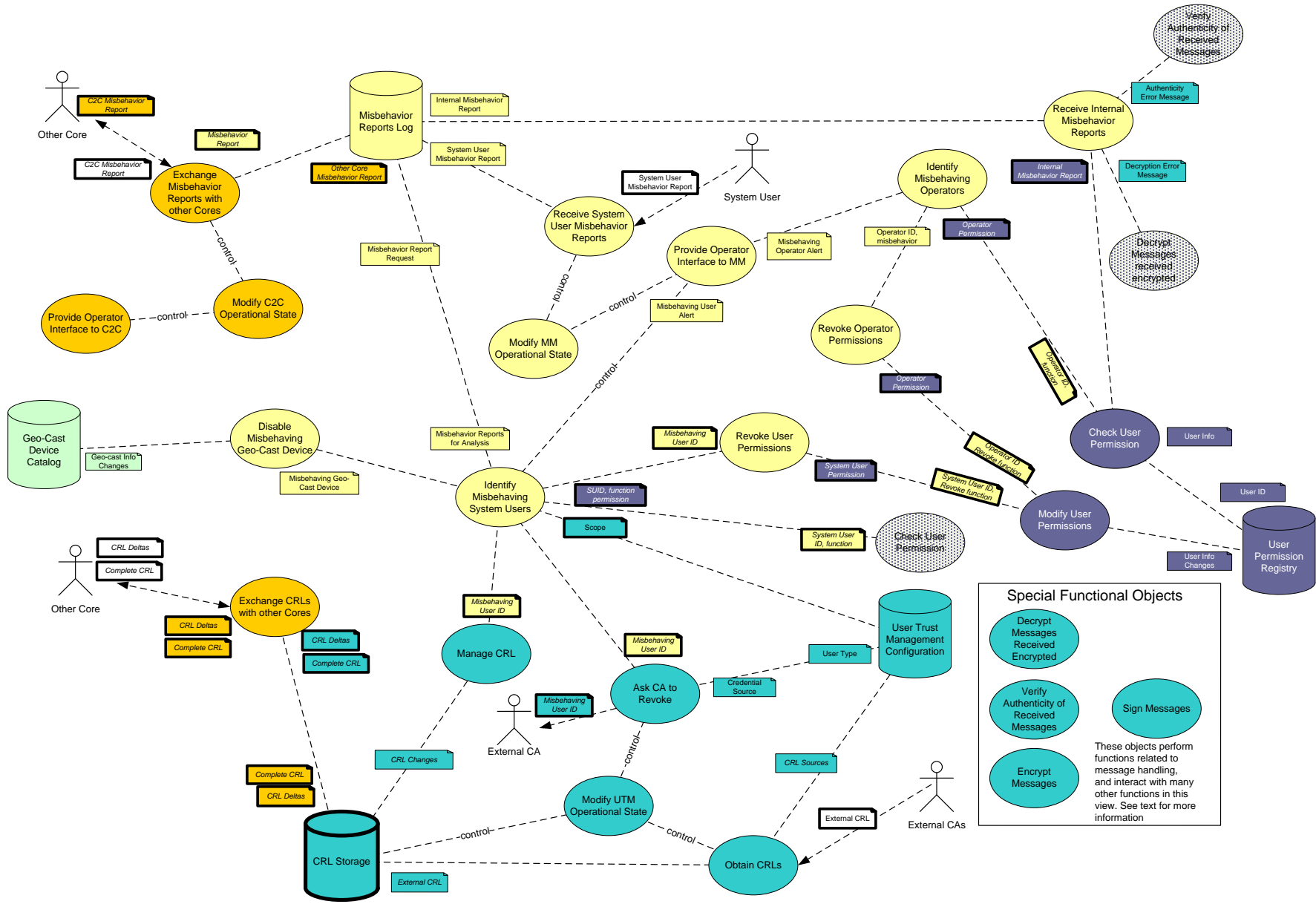
- Description:
 - Identify System Users that are not acting properly
 - Prevent the actions of misbehaving System Users from negatively affecting other System Users
 - Core requests actions of the External CA to deal with the misbehaving user (cert holder)
 - Monitor Operators, and identify and act when they operate in such a way as to jeopardize the Core or the information it passes and stores

Functional View 4.2.7 – Misbehavior Management

■ Considerations/Concerns Addressed:

Interfaces	How does the Core System enable control of the services it provides?
Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Security	What functional elements are involved in the distribution and revocation of digital certificates, and what roles do those entities have? What functional elements are involved in the detection of misbehavior by System Users, and what roles do those entities have?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?

Functional View 4.2.7 – Misbehavior Management



Functional View 4.2.7 – Misbehavior Management

Related Views:

- Enterprise Views
 - Security Credentials
 - Governance
- Functional Views
 - Top Level
 - User Configuration
 - Credentials Distribution
- Connectivity Views
 - Core System Functional Allocation
- Information Views
 - Top Level External Objects
 - Top Level Internal Objects

Functional View 4.2.8 – Core Decryption

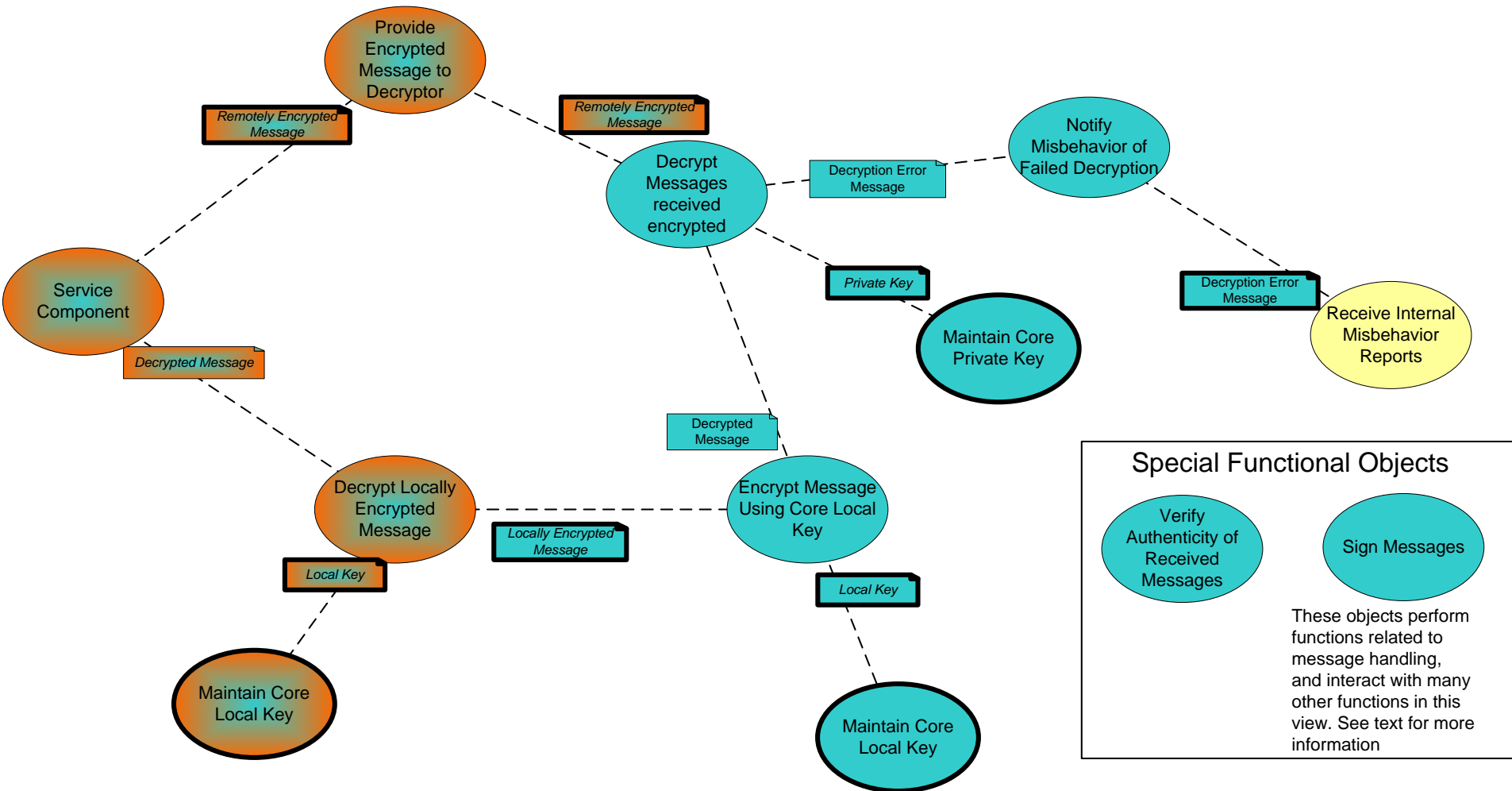
- Description:
 - Encrypted messages meant for the Core must be decrypted in order for the Core to act on their contents
 - *Encrypted messages that come to the Core but are addressed to System Users are **not** decrypted by the Core.*
 - Requires the storage of a private key at the receiving nodes
 - Balance security risk of having key in multiple locations

Functional View 4.2.8 – Core Decryption

- Considerations/Concerns Addressed:

Security	How does the Core System secure System Users' personal information?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?

Functional View 4.2.8 – Core Decryption



Functional View 4.2.8 – Core Decryption

- Alternatives Considered:
 - Store the private encryption key at each node that requires the ability to read encrypted messages directed to the Core
 - Supports scalability
 - Exposes security risks

Functional View 4.2.8 – Core Decryption

Related Views:

- Functional Views
 - Top Level
- Connectivity Views
 - High Level
 - Core System Functional Allocation
- Communications Views
 - Mobile DSRC Device and Core
 - Mobile Wide-Area Wireless User and Core
 - Fixed Point Center/Field User and Core, Core2Core
- Information Views
 - Top Level External Objects
 - Top Level Internal Objects

Functional View 4.2.9 - Networking

- Description:
 - Functionality required to maintain security and provide communications for the Core
 - Addresses the Core's connectivity to private networks and the Internet
 - Defense against attack through those networks
 - All network traffic must be passed through the Intrusion Prevention System (IPS).

Functional View 4.2.9 - Networking

■ Considerations/Concerns Addressed:

Interfaces	How does the Core System enable control of the services it provides?
Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?
Evolvability	How easily can the Core's functionality be expanded to cover new needs if they arise?

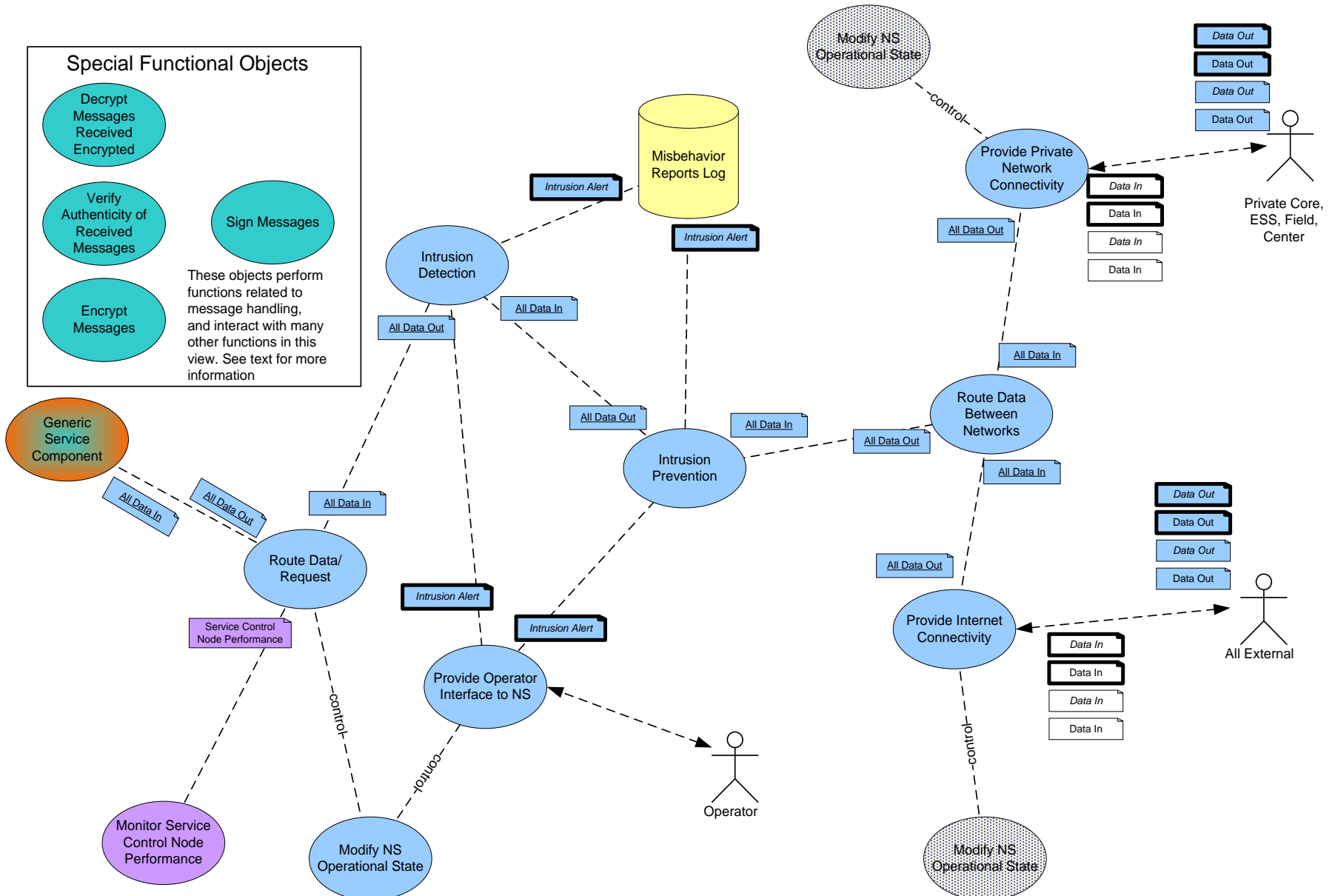


Functional View 4.2.9 - Networking

Special Functional Objects

- Decrypt Messages Received Encrypted
- Verify Authenticity of Received Messages
- Sign Messages
- Encrypt Messages

These objects perform functions related to message handling, and interact with many other functions in this view. See text for more information



Functional View 4.2.9 - Networking

- Alternatives Considered:
 - Network traffic could be allowed in without being passed through a single Intrusion Prevention System (IPS)
 - Opens up a potential bottleneck
 - Relies on separate security systems
 - Considered but rejected to preserve security/integrity of the system

Functional View 4.2.9 - Networking

Related Views:

- Connectivity Views
 - High Level
 - Core System Functional Allocation
- Communications Views
 - Core Routing
- Information Views
 - Top Level External Objects
 - Top Level Internal Objects

Functional View 4.2.10 – Core Backup

- Description:
 - Core Systems may provide backup functionality to one another.
 - Backup of services, where one Core may provide services in behalf of another Core
 - Backup of data, since data backup is required to implement service backup

Functional View 4.2.10 – Core Backup

■ Considerations/Concerns Addressed:

Interfaces	How does the Core System enable control of the services it provides?
Functionality	How does the Core System monitor the services it provides? How does the Core System support the coordination of resources between different Cores? How does the Core System function internally? How do the Core System's components work together? How does the Core System transition between operational modes?
Security	How does the Core System secure System Users' personal information?
Appropriateness	Does the Core System meet all of the needs defined in the ConOps? Does the Core System meet all of the functional requirements defined in the SyRS?

Functional View 4.2.10 – Core Backup

Related Views:




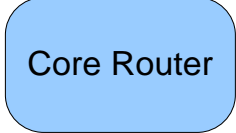
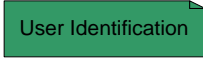

- Enterprise Views
 - Security Credentials Distribution
 - Configuration and Maintenance
 - Operations
 - Governance
 - Business Model Facilitation
- Functional Views
 - Top Level
 - System Configuration
 - Credentials Distribution
- Information Views
 - Top Level External Objects
 - Top Level Internal Objects

Core System Architecture: Connectivity Views

»» Nodes, communications Links,
and Applications

Connectivity

- Composition of the physical elements (nodes) and their connections and interactions
- Links are traceable to *interface* requirements

	Nodes
	link between Nodes, likely a wired connection
	link between Nodes, likely a wireless connection
	Applications external to the Core System and Core Functional Objects
	An Information Object
	Ports

Connectivity View 4.3.1 – High Level

- Description:
 - Allow deployment of the Core System across multiple hardware nodes
 - Provides the interface between the Internet and the Core System's Service Component Node
 - Also supports connectivity through private networks

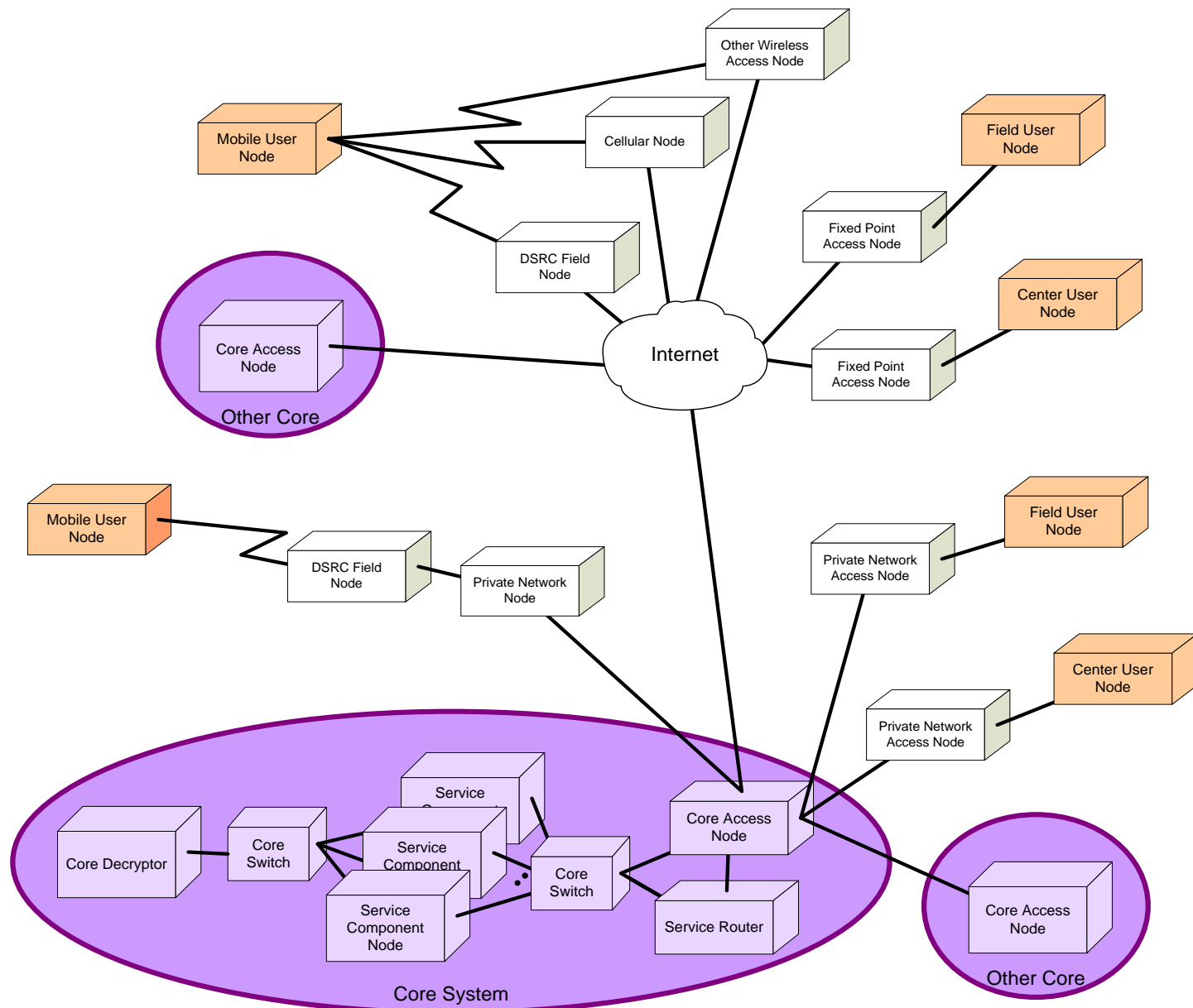
Connectivity View 4.3.1 – High Level

■ Considerations/Concerns Addressed:

Performance	Can the Core meet all of the performance requirements defined in the SyRS (e.g., availability, reliability, capacity, and other quantitative measures)?
Security	How are the Core System components secured from network attack? How are the Core System components physically secured?
Feasibility	Are Core System services feasible to develop given current technology and resources?
Risks	Is the Core System’s hardware and software architecture susceptible to failure, and if so under what circumstances? What are the characteristics of this failure?
Evolvability	Is the structure of the Core System sufficiently flexible and scalable to deploy and to enable changes to cover new needs if they arise?
Maintainability	Can the Core’s functionality be sustained with acceptable levels of downtime as per the SyRS?



Connectivity View 4.3.1 – High Level



Connectivity View 4.3.1 – High Level

- Alternatives Considered:
 - Whether to allow private networks to connect to Core Systems was reviewed in June vs. requiring all System Users to come through the Internet
 - Decided to allow private network connections
 - Greater flexibility, promote deployment

Connectivity View 4.3.1 – High Level

Related Views:

- Enterprise Views
 - Configuration and Maintenance
 - Governance
 - Business Model Facilitation
- Functional Views
 - Core Decryption
 - Networking
- Connectivity Views
 - Core System Functional Allocation
- Communications Views
 - Mobile DSRC Device and Core
 - Mobile Wide-Area Wireless User and Core
 - Fixed Point Center/Field User and Core, Core2Core
 - Core Routing

Connectivity View 4.3.2 – Core System Functional Allocation

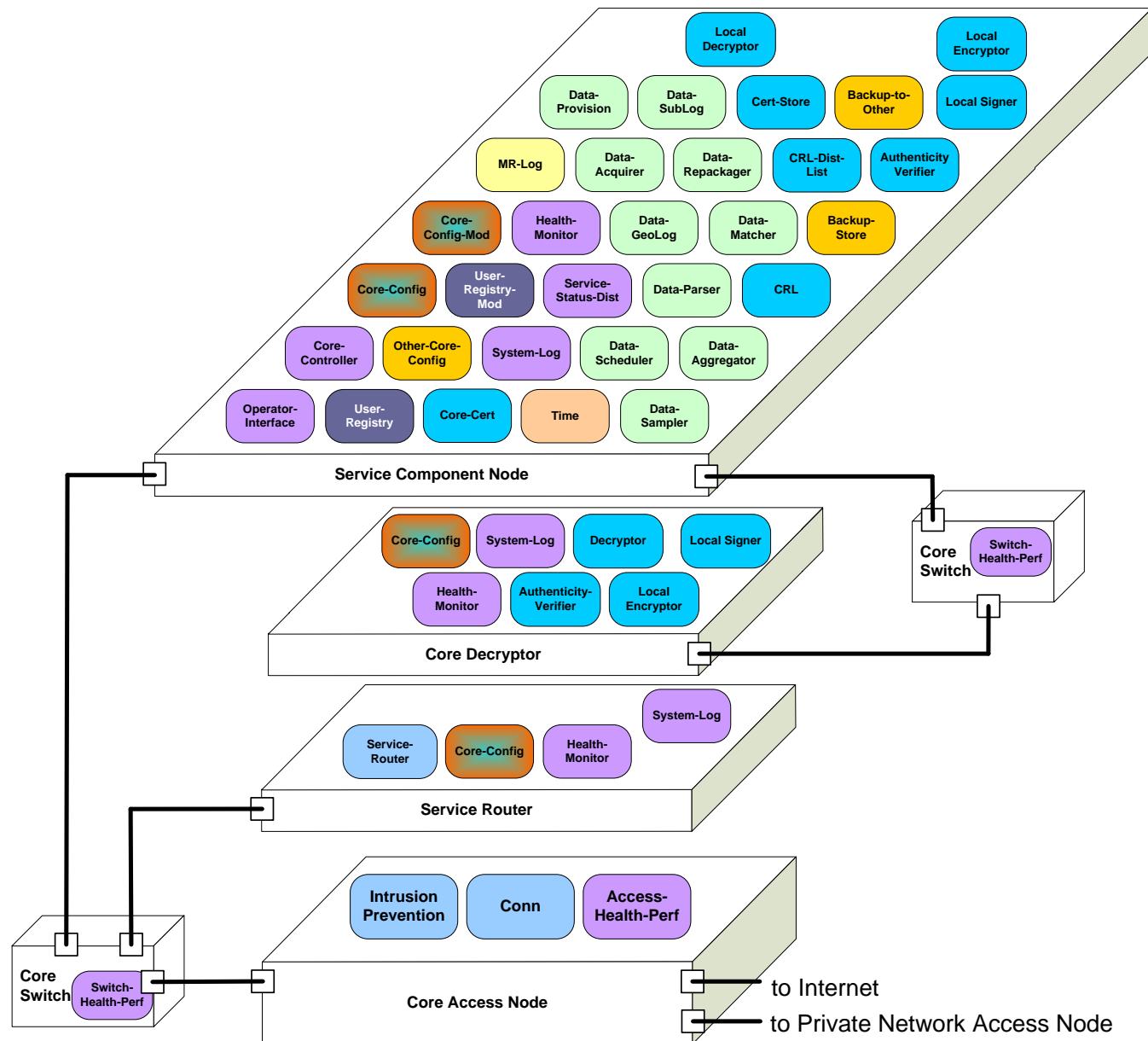
- Description:
 - Allocates functional objects to engineering objects
 - Identifies devices (hardware engineering objects, nodes) and Software Engineering Objects (SEOs)
 - Possible to vary how SEOs are distributed among nodes
 - Supports redundancy

Connectivity View 4.3.2 – Core System Functional Allocation

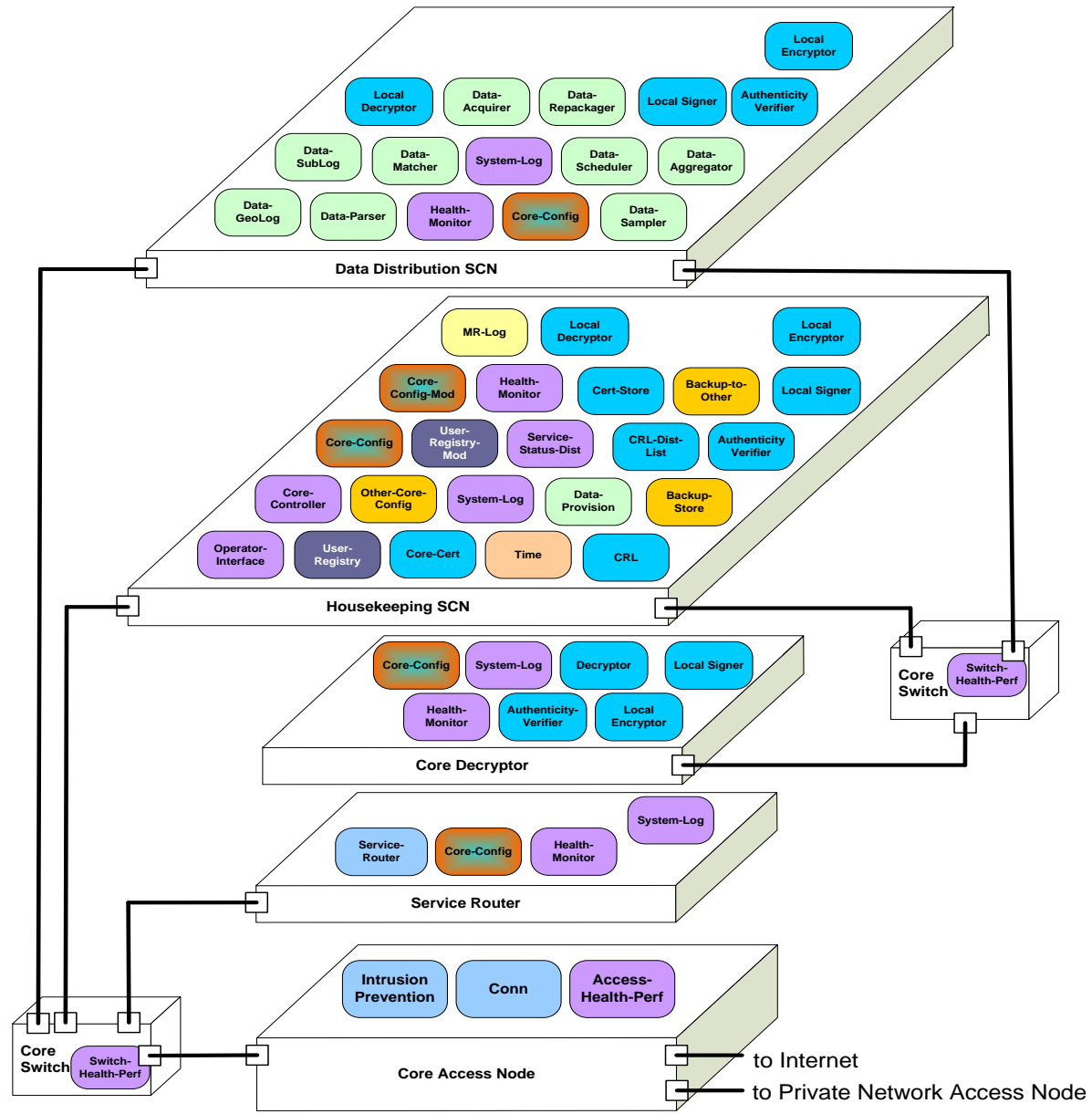
- Considerations/Concerns Addressed:
 - Performance
 - Interfaces
 - Security
 - Feasibility
 - Risks
 - Evolvability
 - Deployability
 - Maintainability



Connectivity View 4.3.2 – Core System Functional Allocation



Connectivity View 4.3.2 – Core System Functional Allocation, variation



Connectivity View 4.3.2 – Core System Functional Allocation

- Other variations to the above shown in SAD
- Alternatives Considered:
 - Allow SCNs, Core Decryptor and Service Router to be implemented on Nodes that are not connected to the same LAN
 - Allowed as an option
 - Put Core Decryptor on same LAN as other components – simplifies network configuration but could compromise security
 - Rejected
- Related to all functional and comm views

Connectivity View 4.3.3 – State and Mode Transitions

- Description:
 - Covers the states and modes and transitions for the hardware and software objects
 - Standby or Operational State
 - Normal, Degraded, Restricted, Maintenance, Degraded/Restricted
 - Training or Installation State
 - Normal, Degraded, Maintenance

Connectivity View 4.3.3 – State and Mode Transitions

■ Considerations/Concerns Addressed:

Performance	Can the Core meet all of the performance requirements defined in the SyRS (e.g., availability, reliability, capacity and other quantitative measures)?
Risks	Is the Core System's hardware and software architecture susceptible to failure, and if so under what circumstances? What are the characteristics of this failure?
Maintainability	Is the structure of the Core System maintainable with a reasonable allocation of resources for the entities that are likely to consider deployment? Can the Core's functionality be sustained with acceptable levels of downtime as per the SyRS?

■ Related View: Functional – Top Level

Connectivity View 4.3.3 – State and Mode Transitions

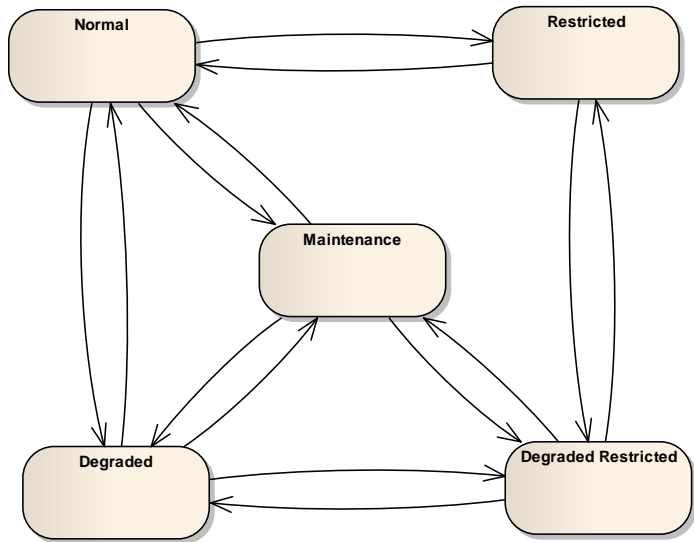


Figure 4-23: Standby and Operational

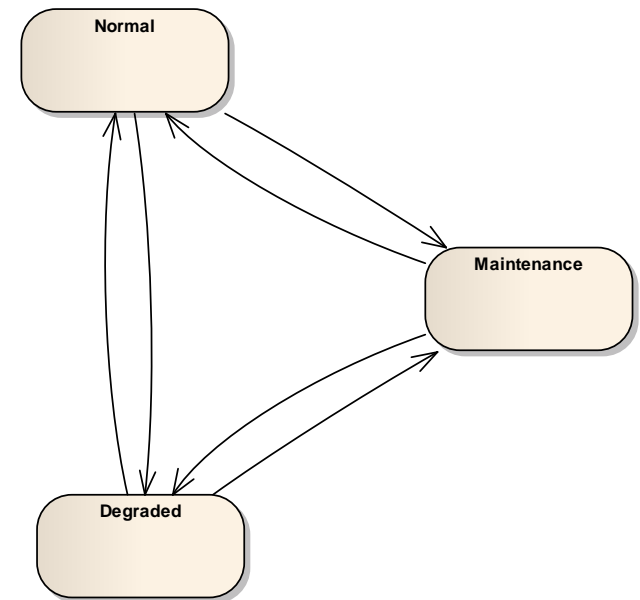


Figure 4-24: Training and Installation Modes

End of Day 2

- »» Return tomorrow for discussion on Communications and Information Views and Other Topics