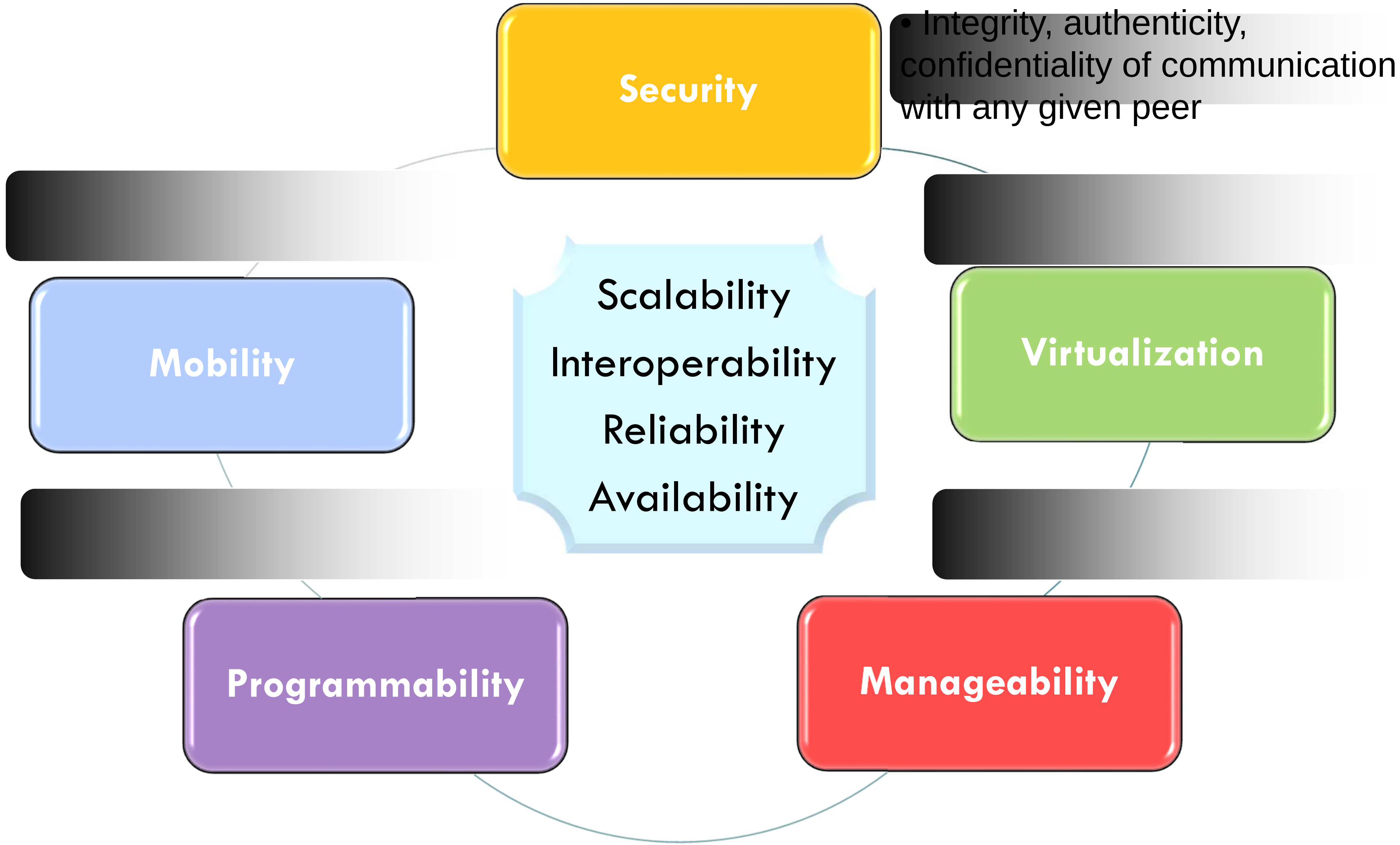
Manageability of Future Internet

Choong Seon Hong
Kyung Hee University
cshong@khu.ac.kr
June 4, 2013

Contents

- Introduction to Future Internet and its Manageability
- GENI Working Groups related to Mgmt
- GMOC
- Federation

Requirements for Future Internet

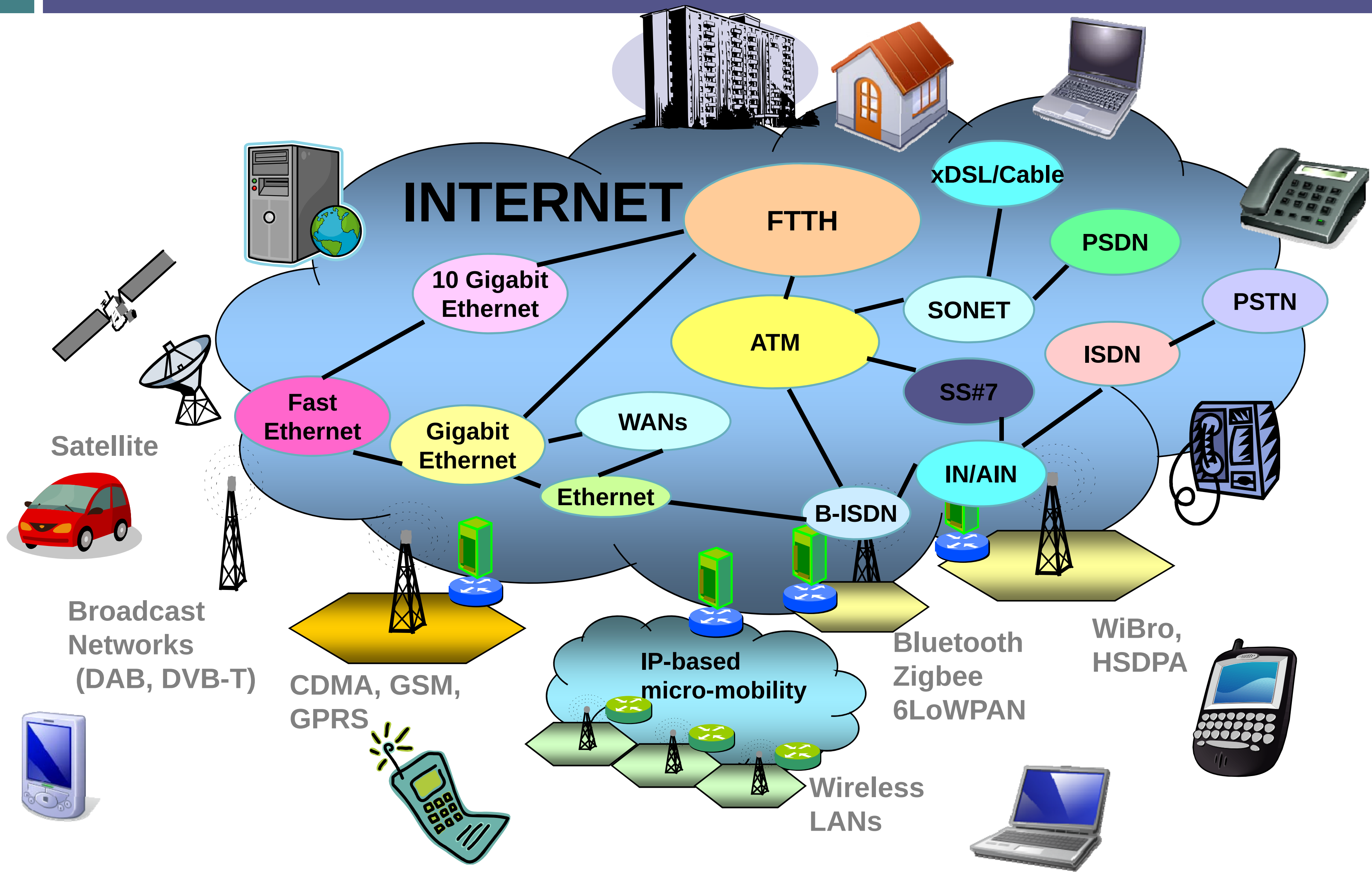


Requirements for Future Internet

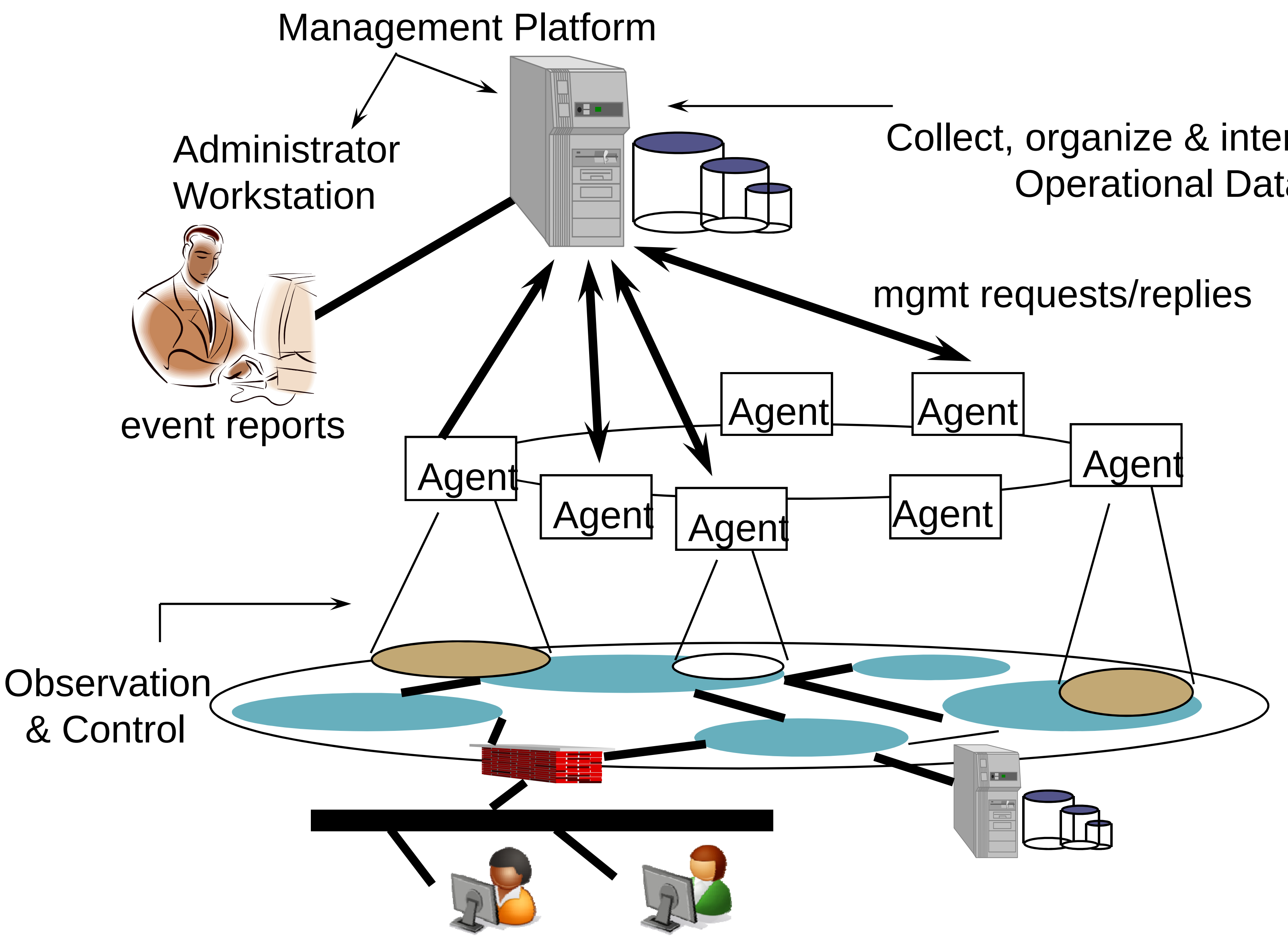
- Integrity, authenticity, confidentiality of communication with any given peer

Management of the Current Internet

Current Network Environment



Current Network Management Framework



Functional Requirements for NM

- Fault Management
  - detection, isolation and correction of abnormal operations
- Configuration Management
  - identify managed resources and their connectivity, discovery
- Accounting Management
  - keep track of usage for charging
- Performance Management
  - monitor and evaluate the behavior of managed resources
- Security Management
  - allow only authorized access and control

FCAPS

Standard Management Frameworks

- OSI Network Management Framework
  - CMIP (X.700 Series)
- Internet Network Management Framework
  - SNMPv1
  - SNMPv2
  - SNMPv3
- TeleManagement Forum
  - SID, eTOM, NGOSS
- Distributed Management Task Force
  - CIM, WBEM
- Open Mobile Alliance
  - OMA DM

Towards Management of the Future Internet

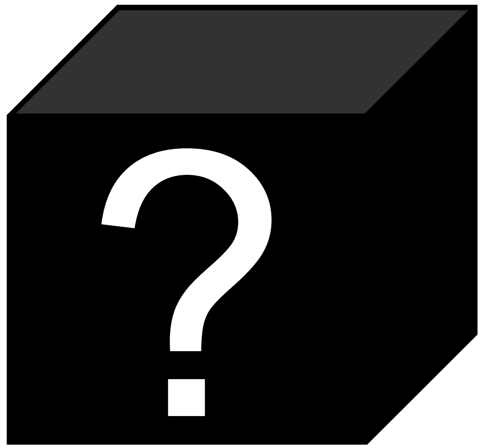
Towards Management of the Future Internet

Manageability for the current Internet has been developed as an afterthought!

THINK about Manageability of Future Internet

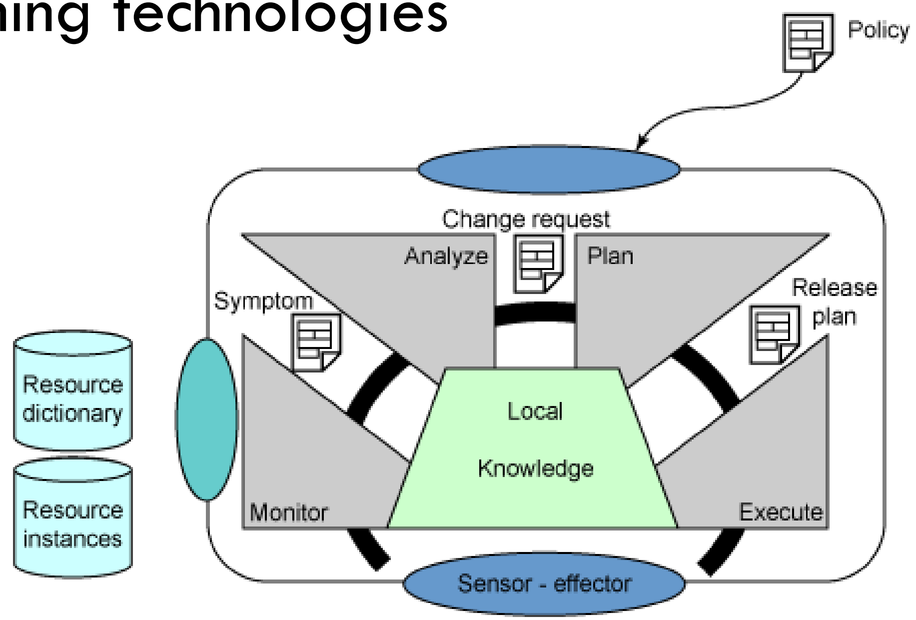
Do we need a revolutionary approach or an evolutionary approach?

FCAPS +



Management for Future Internet

- Autonomic Management/Self-Management
  - Self-managing frameworks and architecture
  - Knowledge engineering, including information modeling and ontology design
  - Policy analysis and modeling
  - Semantic analysis and reasoning technologies
  - Virtualization of resources
  - Orchestration techniques
  - Self-managed networks
  - Context-awareness
  - Adaptive management



Research Efforts for Management of FI

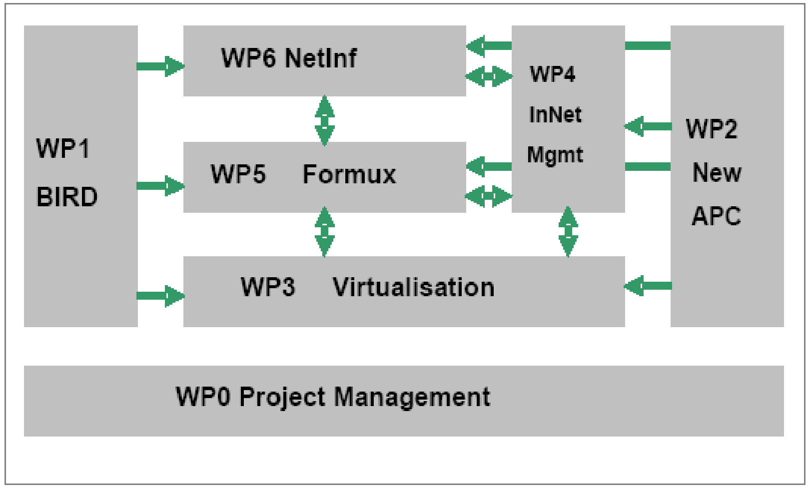
- US NSF
  - Future Internet Design (FIND)
    - Complexity Oblivious Network Management architecture (CONMan)
  - Global Environment for Networking Innovations (GENI)
    - Operations, Management, Integration and Security (OMIS) WG
- EU
  - Framework Program (FP) 7
    - 4WARD In-network (INM) project
    - Autonomic Internet (AutoI) project
    - Autonomic Network Architecture (ANA) project

CONMan: Overview

- Management interface should contain as little protocol-specific information as possible
- Complexities of protocols should be masked from management
- Goal
  - A generic abstraction of network entities (protocols & devices) for management purpose
  - A set of atomic management operations to work upon the abstraction
  - A way to translate high-level management objectives to low-level operations

Research Efforts - EU

http://www.4ward-project.eu



- 4WARD WP4: INM (In Network Management)
  - Autonomic self-management
  - Abstractions and a framework for a self-organizing management plane
  - Scheme, strategies, and protocols for collaborative monitoring, self-optimizing, and self-healing

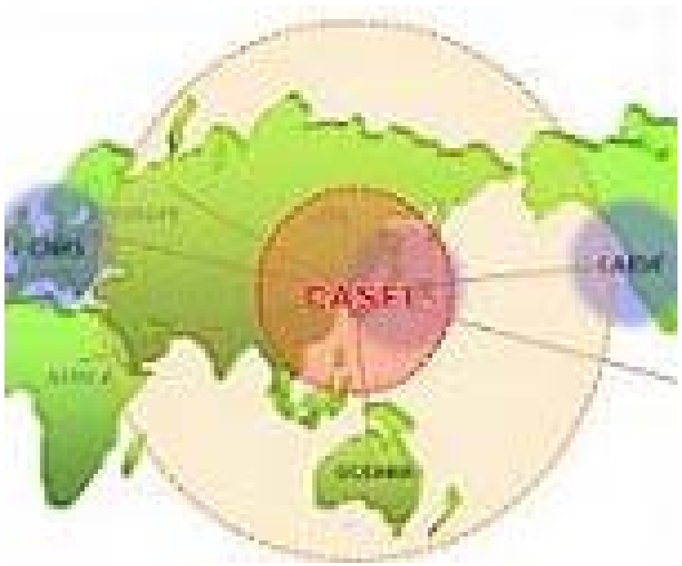
Research Efforts - USA

- GENI OMIS WG (**O**perations, **M**anagement, **I**ntegration and **S**ecurity)
  - Operations, management, integration and security processes in GENI
  - Experiment support, monitoring, and data storage
  - Security monitoring and incident response
  - Federation management and monitoring
  - Hardware release, maintenance and integration
  - Software release, maintenance and integration
  - Operations metric collection and analysis

  - http://www.geni.net/wg/omis-wg.html

Research Efforts - Korea

- **CASFI** (**C**ollect, **A**nalyze, and **S**hare for **F**uture **I**nternet)
  - Goals
    - **Manageability of Future Internet**
    - Data Sharing Platform for Performance Measurement
    - High-Precision Measurement and Analysis
    - Human Behavior Analysis
  - Groups
    - KHU, KAIST, POSTECH, CNU
  - Period
    - 2008.03.01 ~ 2013.02.28
  - http://casfi.kaist.ac.kr

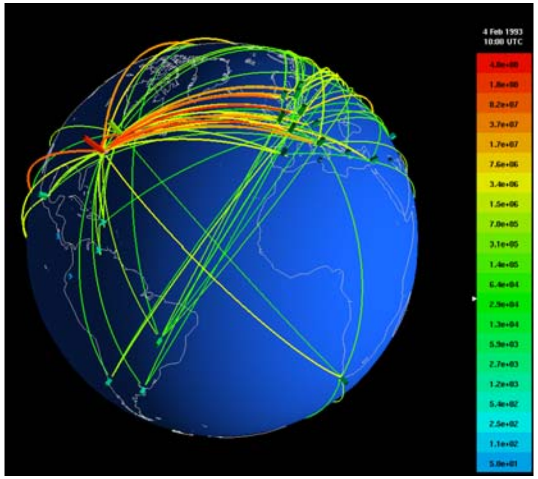


Management for Future Internet [1]

- Management Interface
  - Management Information Modeling & Operations
  - Instrumentation
- Management Architecture
  - Centralized vs. Decentralized Management
  - Peer-to-Peer
  - Hybrid
- Service Management
  - Customer-centric service
  - Service portability
  - SLA/QoS

Management for Future Internet [2]

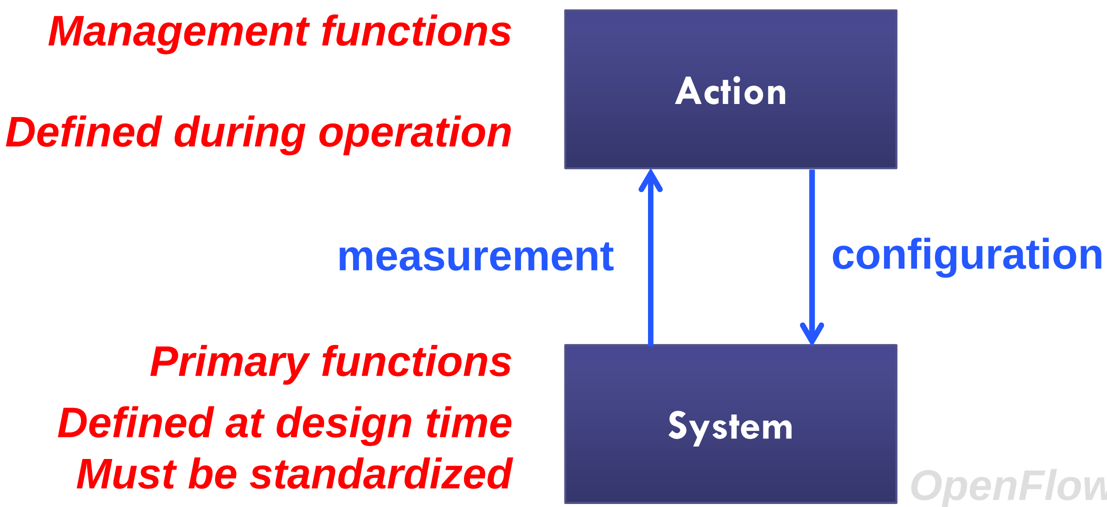
- Traffic Monitoring/Measurement and Analysis
  - Monitoring for large-scale and high-speed networks
  - Network/application-level monitoring
  - Global traffic data access/sharing
  - Fast and real time monitoring
  - Statistical sampling method
  - Storing method for large scale traffic data
  - Measurement and analysis of social networking



Network Management Architecture

1. What is Management?

Network management is the act of initializing, monitoring and modifying the operation of the primary functions



OpenFlow enables networks to evolve, by giving a remote controller the power to modify the behavior of network devices, through a well-defined "forwarding instruction set"

2. Why Management?

- Cost reduction
- Flexibility
- Lack of Experience
- Fault handling
- Security

Cost Reduction

- General purpose designs
  - Internet, VoIP, SCADA, Server Farms, Internet of Things, …

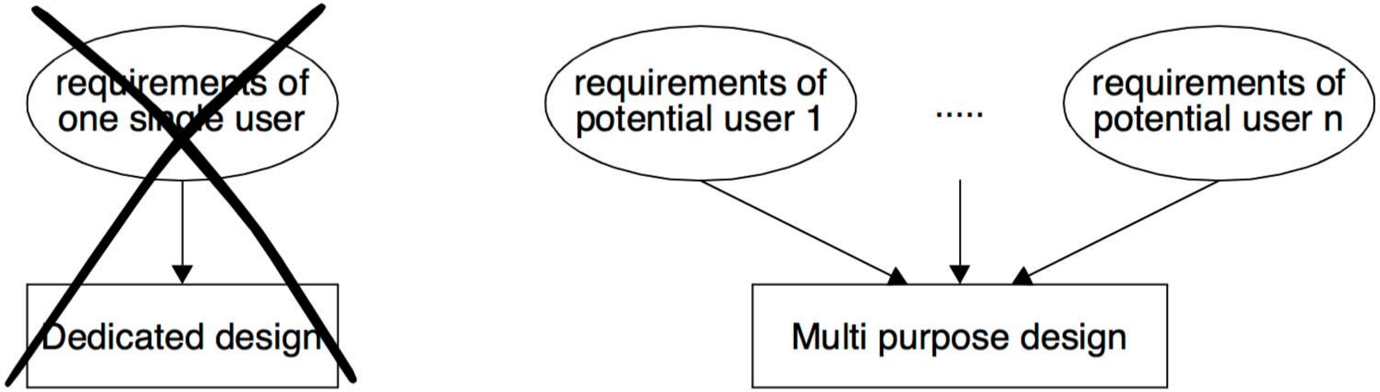


Figure 1.2: Design should not be customized, but general purpose

Flexibility

- Changing user requirements

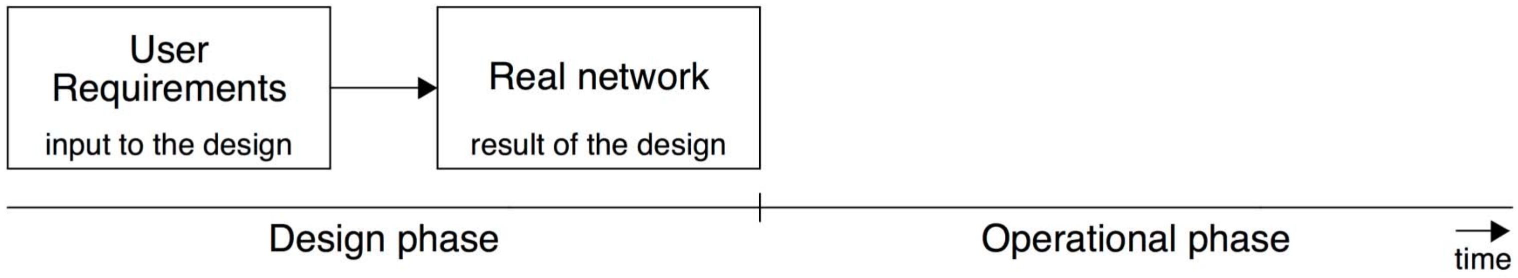


Figure 1.4: Simplified top-down design process

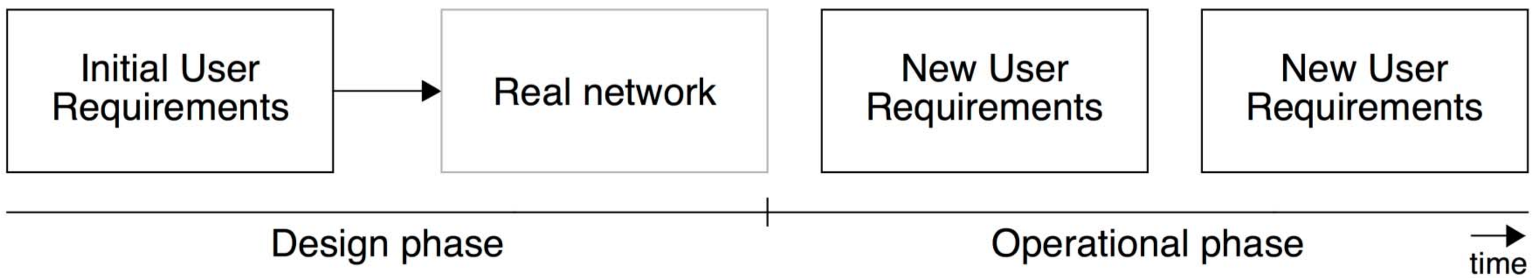


Figure 1.5: Changing user requirements

Flexibility: Cyclic design

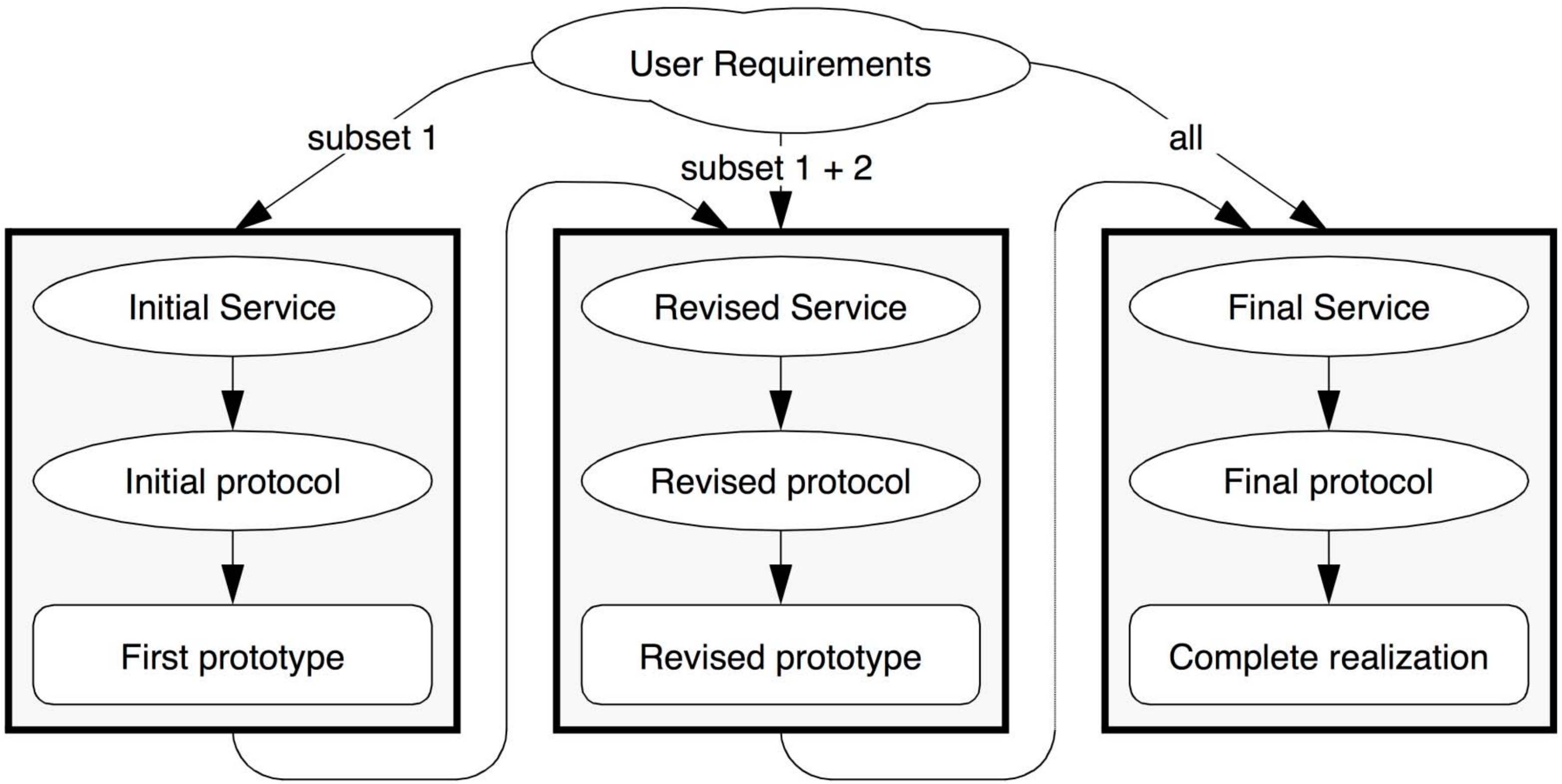


Figure 6.1: Cyclic design

3. How is Management performed?

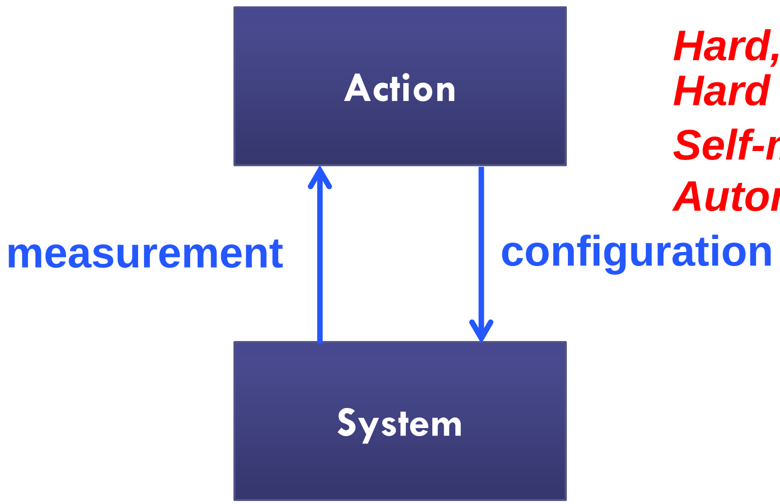
a. Explicit versus implicit management

b. Centralized versus distributed management

c. Meta-Management

3.B Explicit versus Implicit

- Explicit management
- Implicit management



Hard, Soft AND Brainware
Hard and Software
Self-management
Autonomic networking

From explicit to implicit management

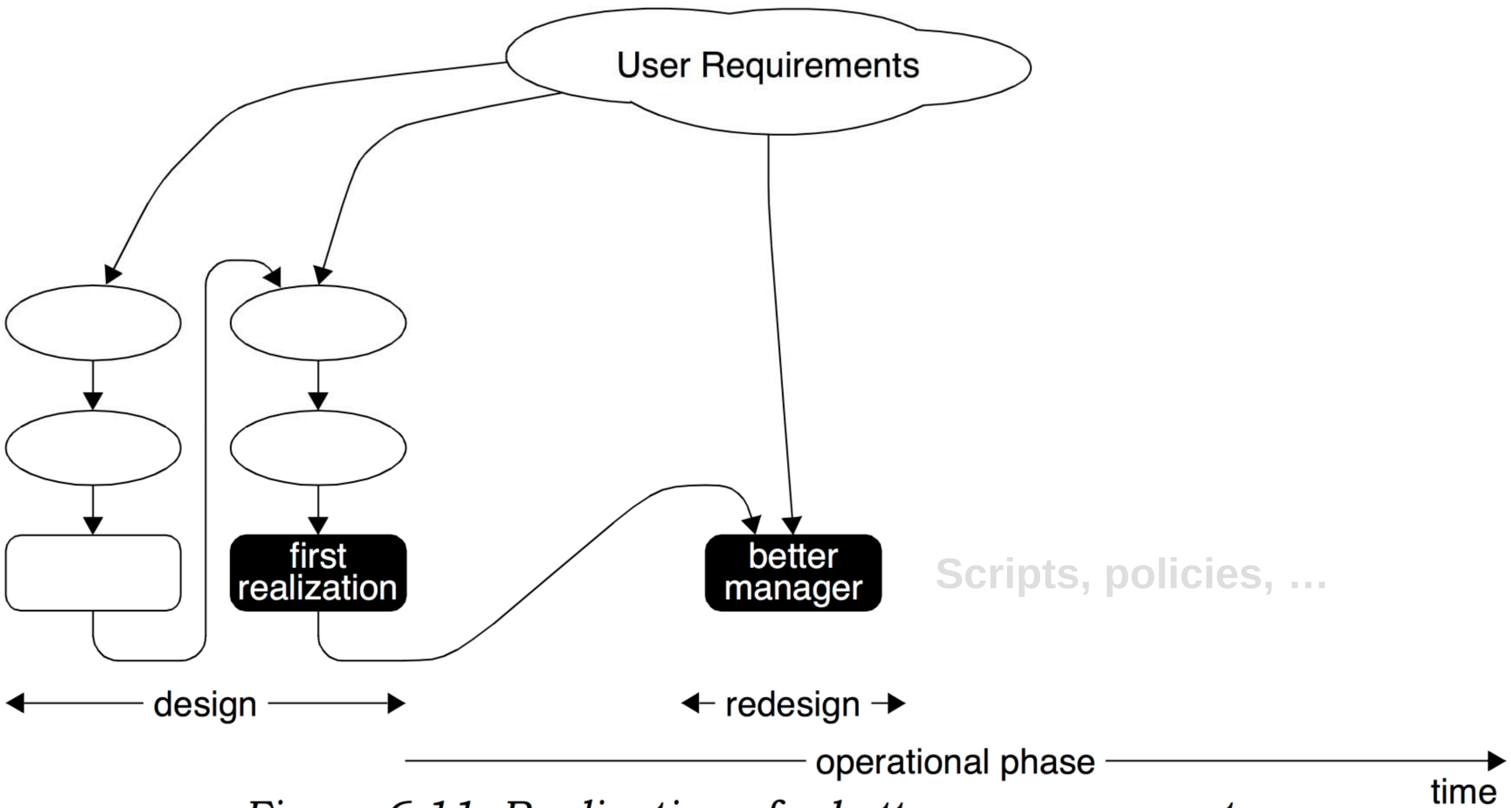


Figure 6.11: Realization of a better manager system

From explicit to implicit management

- *Management needs to be increasingly part of the functionality of a managed object, not something which can be added afterwards*
- *Future Networks will challenge Service and Network Operators to find the right ways to embed intelligence into networks in order to ensure their autonomic management and control*

3.b Centralized versus Distributed

- Centralized
  - DNS
  - DHCP
  - SNMP
  - NetConf
- Distributed
  - ZeroConf, Bonjour, …

From centralized to distributed management

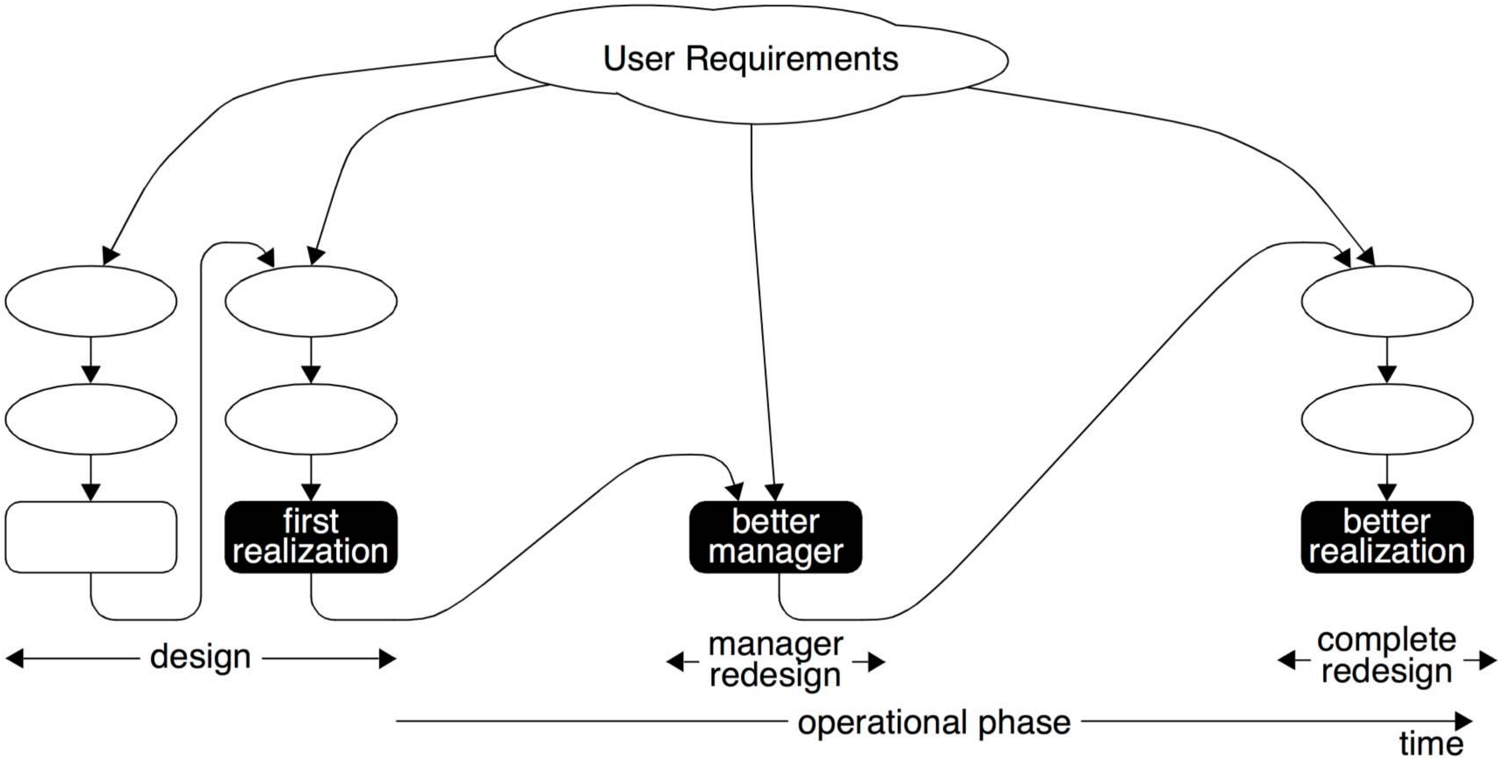
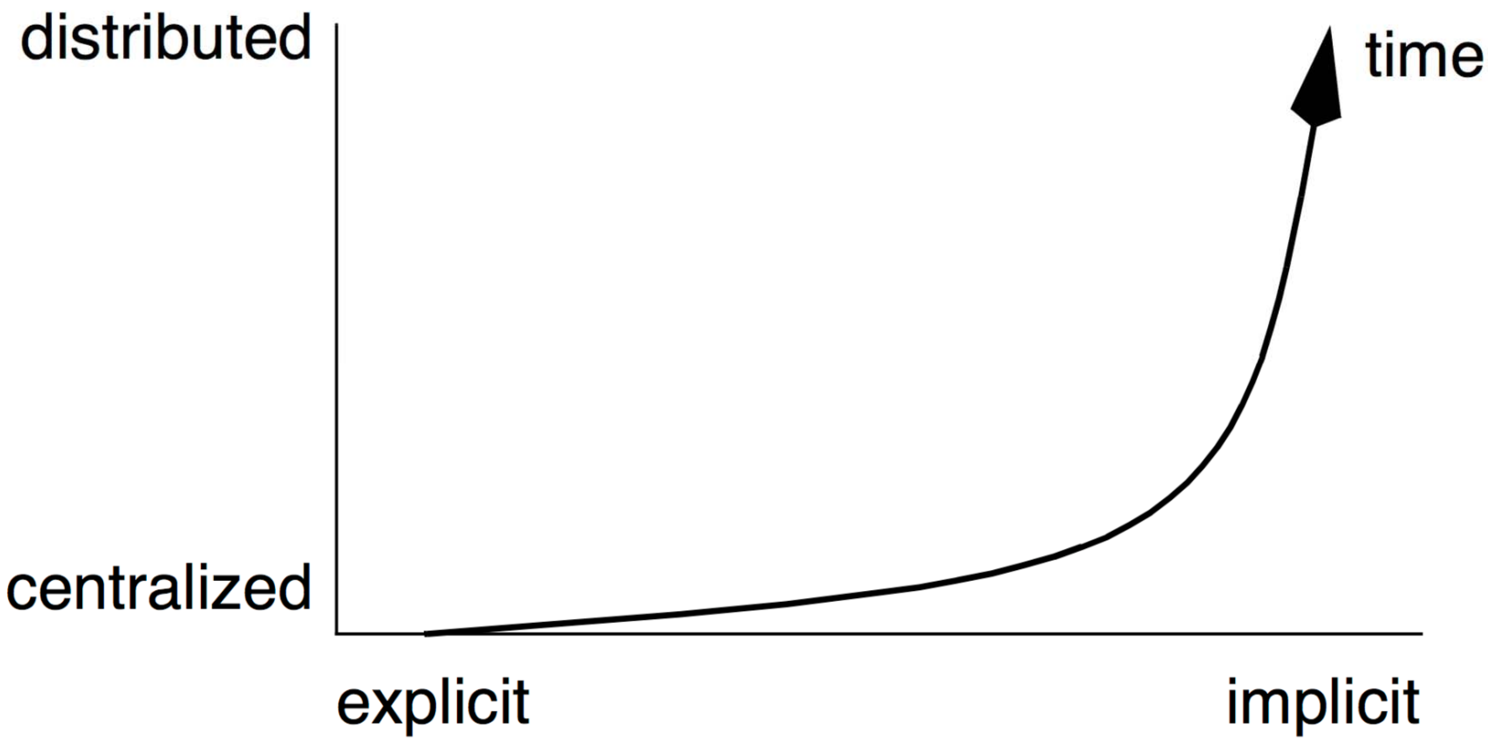


Figure 6.14: Additional design cycle to distribute management functions

From explicit and centralized to implicit and distributed management



3.c Meta Management

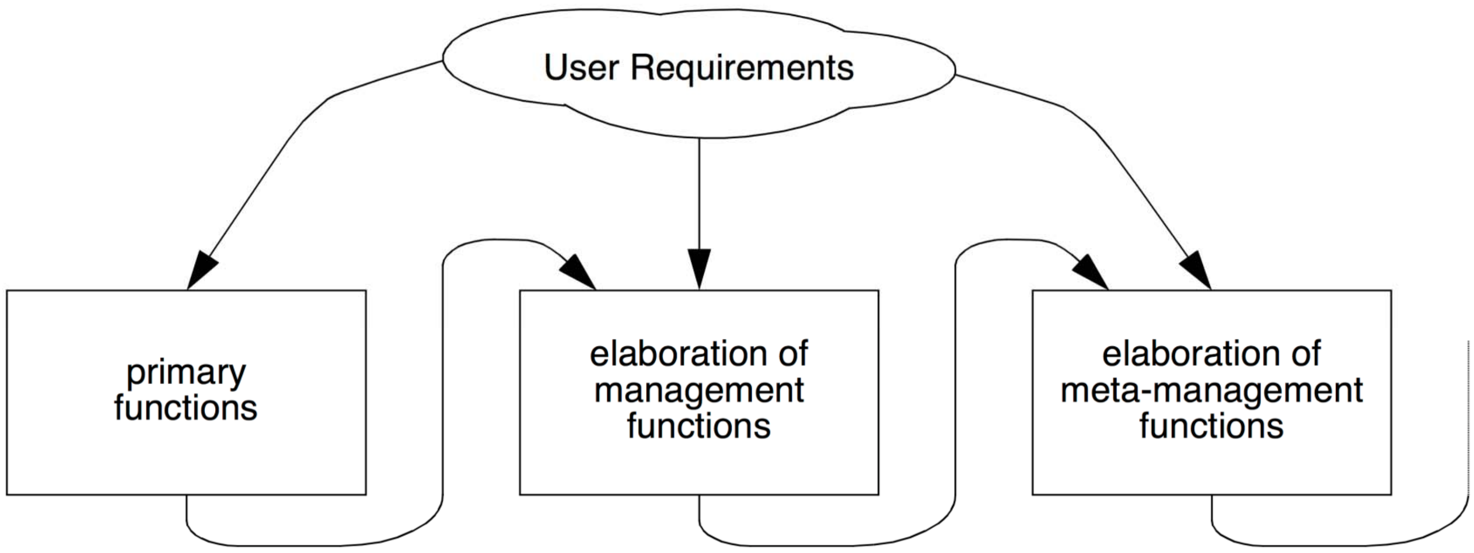


Figure 6.8: The addition of meta-management to the design

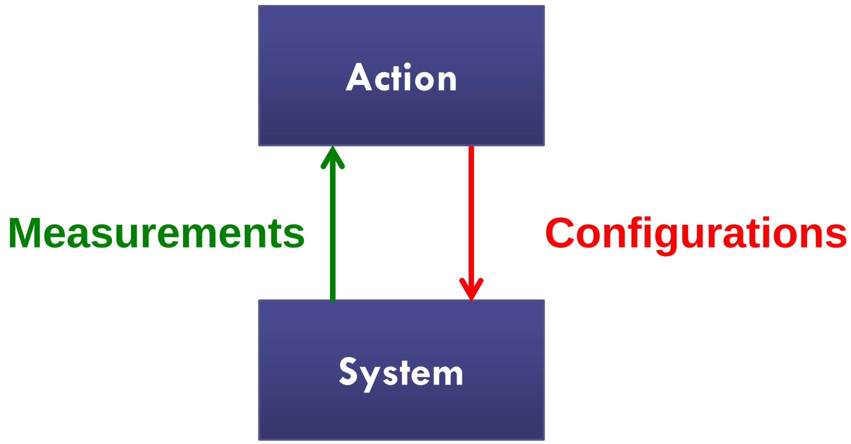
Management is a moving target: in later design cycles we will have to manage the management functions from the previous cycles (meta-management)

Predicting the future: conclusions

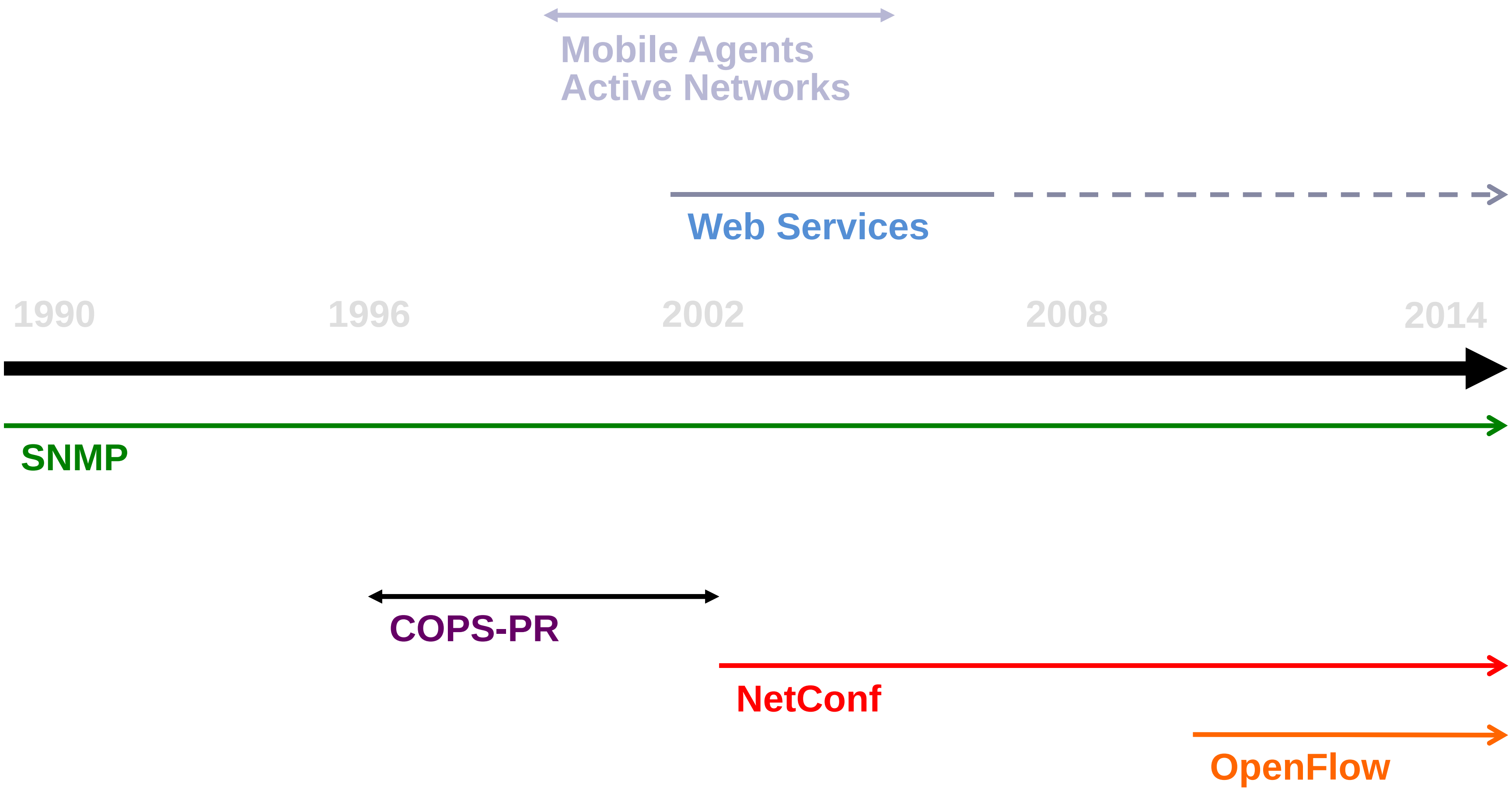
- There are several management invariants
  - The reasons to perform management (WHY) do not change
  - The way we do management (HOW) remains relatively stable
    - From explicit to implicit
    - From centralized to distributed
  - The management functions we have to design are a moving target
    - Meta-management
- These invariants *may* help determining future directions

Overview

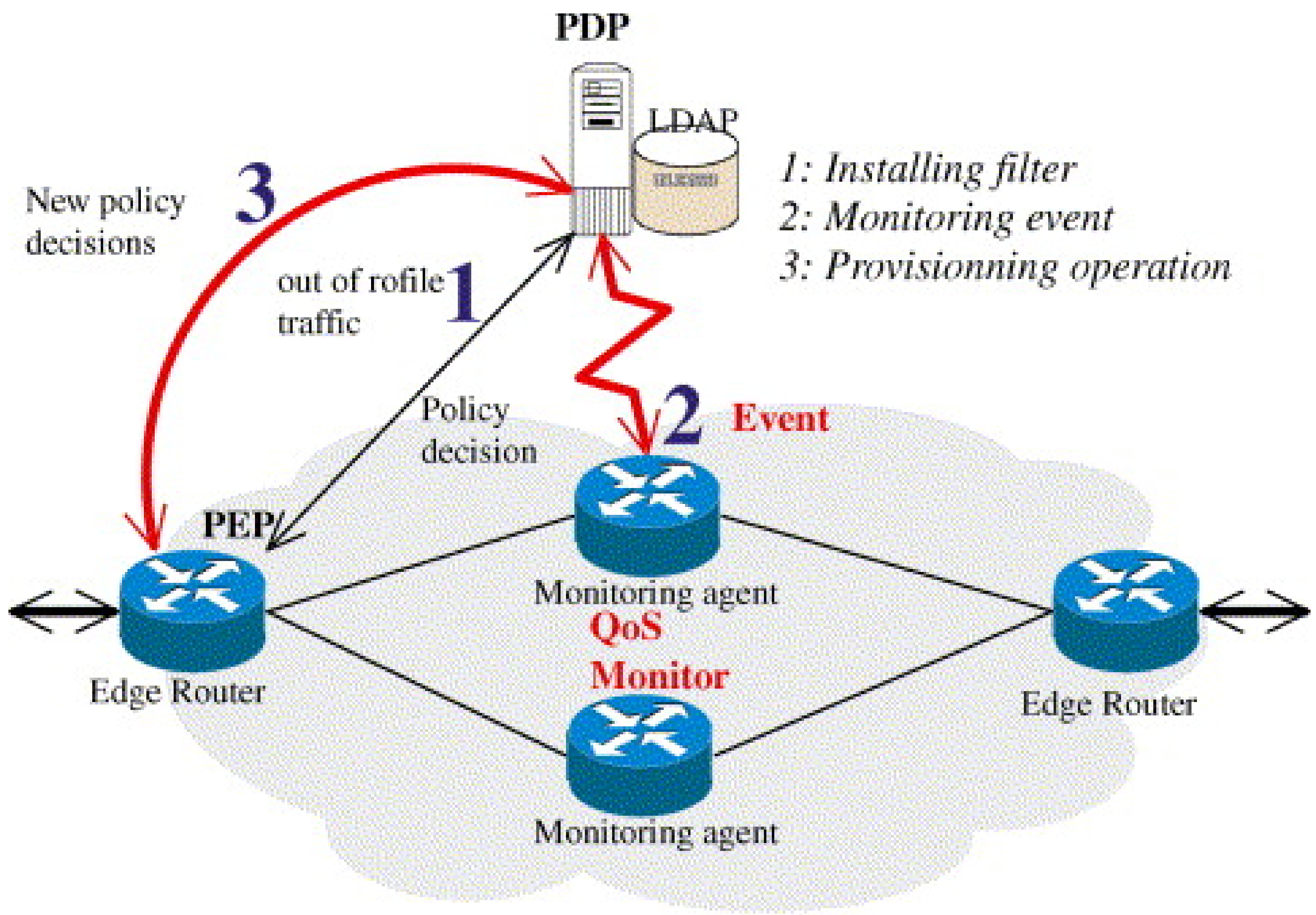
- Network Configuration
  - OpenFlow
  - Relation to SNMP, COPS-PR, NetConf



Timeline


Mobile Agents
Active Networks
Web Services
1990
1996
2002
2008
2014
SNMP
COPS-PR
NetConf
OpenFlow

COPS-PR



PEP: Policy Enforcement Point
PDP: Policy Decision Point

NetConf motivation

- Managers need better control over routing
- Routing protocols such as OSPF lack flexibility
- Routing decisions should be made at a central site

Requirements of Internet Network Operators

The Operations and Management Area of the IETF organized several meetings in 2001 to identify and outline a set of requirements for Internet network operators in order for management protocol and application developers to better meet their needs. In June 2002 the IAB organized a workshop on the configuration aspects of network management [3].

During these meetings, it became clear that, from the operators' standpoint, configuration management is the most important problem to be addressed to date. Operators of large backbone networks maintain their network-wide configuration data in a logically centralized database, as depicted in Fig. 1 [4]. Change requests leading to configuration changes in network devices (e.g., new routing policies) trigger transactions on the logically centralized database. Once a new network-wide configuration has been established in the database, complete configuration files or incremental configuration updates for specific network devices are first generated by a configuration data translator, then distributed to all devices, and finally activated. It is not unusual for Internet network operators to write these translators themselves. Due to a lack of well established standards, network operators have to update their translators when new network devices are released, or when new firmware needs to be installed in already deployed devices.

The requirements of the Internet network operators can be summarized as follows:
- It is crucial to make a clear distinction between configuration data (which is rather static) and data that describes operational state (which is dynamic by nature).
- There must be basic operations to download and upload complete configuration files. It is desirable to be able to download or upload only parts of the configuration data.
- The configuration data should be in a textual format to allow the usage of a wide range of text-processing tools (e.g., the UNIX command diff) and version management systems.
- It is necessary to distinguish between the distribution of configurations and the activation of a certain configuration. Devices should be able to hold multiple configurations and enable management applications to activate any of them (only one configuration is active at a time).
- The coordinated activation of configurations could be dramatically simplified by having a transaction mechanism for uploading new configurations and activating them "simultaneously" on multiple devices. Such a transaction mechanism must take into account that connectivity might be lost in the middle of the transaction.
- Finally, ease of use of the management technology is of paramount importance. Configuration management interfaces must be designed such that developing and debugging configuration data translators is cost effective.

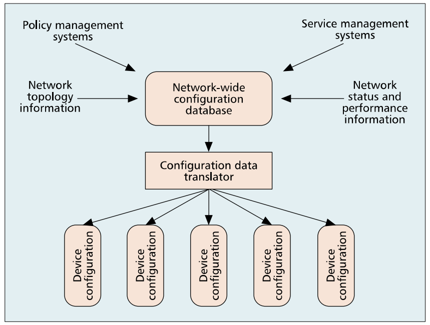


Figure 1. *A configuration management model.*

Management Environment

The SNMP framework was designed to:
- Minimize the number and complexity of management functions realized by the agents
- Be extensible to accommodate additional and unanticipated aspects of network operation and management
- Be as much as possible independent of the implementation of particular hosts or gateways [5]

As a result, the main strengths of SNMP are its simplicity, interoperability, and low footprint on agents [6].

SNMP must also work effectively when the network is not fully operational. This reflects in the selection of a connectionless transport protocol (UDP), which allows management applications to exercise full control over the retransmission strategy.

Another design choice was to keep SNMP as independent as possible of other network services. This is one of the main reasons why, in SNMP version 3 (SNMPv3), security is self-contained and does not rely on other external security services such as key exchange or certification services.

But the environment in which management operations take place has dramatically changed since SNMP was devised. Looking at today's network technologies and the actual usage patterns of SNMP, it is obvious that devices could perform more complex management operations at low cost. It is reasonable to expect that devices, especially high-end routers and switches, will become increasingly programmable, and that it will become possible to execute more control software directly on the devices.

Furthermore, as described by Wellens and Auerbach [7], SNMP need not use UDP. When network connectivity is lost, non-SNMP mechanisms are usually used to bring back connectivity before management operations can resume.

Finally, SNMP was standardized at a time

ComMag 2003

NetConf characteristics

- Intended for Configuration Management
  - Based on XML technology
  - Operates on documents, instead of objects
- Granularity level is therefore high
  - Data models not (yet?) defined
- Security is provided at lower layers
  - Use of TCP
  - use of existing security mechanisms (SSH, TLS, SOAP, BEEP)
- Multiple operations are defined

NetConf - Operations

- Get-Config (Source, Filter)
- Edit-Config(Target, Options, Config)
- Copy-Config(Source, Target)
- Delete-Config(Target)
- Get(Filter)
- Validate(Source)
- Lock(Source)
- Unlock(Source)
- Commit(Confirmed, Confirmed-Timeout)

NetConf – Separation between PDUs and Data

Layers	Example
Content	XML Configuration data
PDU Operations	<get-config>, <edit-config>
RPC	<rpc>, <rpc-reply>
Transport	SSH, HTTPS(TLS), BEEP

Configuration data:
<running> configuration
<startup> configuration
<candidate> configuration

OpenFlow motivation

- Managers need better control over routing
- Routing protocols such as OSPF lack flexibility
- Routing decisions should be made at a central site
- Forwarding hardware and routing software should be decoupled

Requirements of Internet Network Operators

The Operations and Management Area of the IETF organized several meetings in 2001 to identify and outline a set of requirements for Internet network operators in order for management protocol and application developers to better meet their needs. In June 2002 the IAB organized a workshop on the configuration aspects of network management [3].

During these meetings, it became clear that, from the operators' standpoint, configuration management is the most important problem to be addressed to date. Operators of large backbone networks maintain their network-wide configuration data in a logically centralized database, as depicted in Fig. 1 [4]. Change requests leading to configuration changes in network devices (e.g., new routing policies) trigger transactions on the logically centralized database. Once a new network-wide configuration has been established in the database, complete configuration files or incremental configuration updates for specific network devices are first generated by a configuration data translator, then distributed to all devices, and finally activated. It is not unusual for Internet network operators to write these translators themselves. Due to a lack of well established standards, network operators have to update their translators when new network devices are released, or when new firmware needs to be installed in already deployed devices.

The requirements of the Internet network operators can be summarized as follows:

- It is crucial to make a clear distinction between configuration data (which is rather static) and data that describes operational state (which is dynamic by nature).
- There must be basic operations to download and upload complete configuration files. It is desirable to be able to download or upload only parts of the configuration data.
- The configuration data should be in a textual format to allow the usage of a wide range of text-processing tools (e.g., the UNIX command diff) and version management systems.
- It is necessary to distinguish between the distribution of configurations and the activation of a certain configuration. Devices should be able to hold multiple configurations and enable management applications to activate any of them (only one configuration is active at a time).
- The coordinated activation of configurations could be dramatically simplified by having a transaction mechanism for uploading new configurations and activating them "simultaneously" on multiple devices. Such a transaction mechanism must take into account that connectivity might be lost in the middle of the transaction.
- Finally, ease of use of the management technology is of paramount importance. Configuration management interfaces must be designed such that developing and debugging configuration data translators is cost effective.

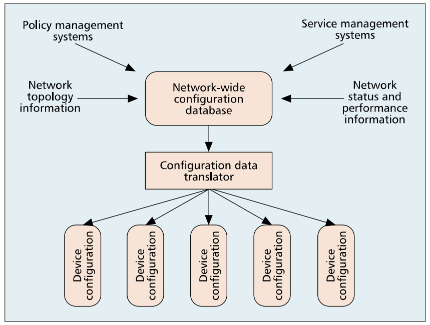


Figure 1. *A configuration management model.*

Management Environment

The SNMP framework was designed to:

- Minimize the number and complexity of management functions realized by the agents
- Be extensible to accommodate additional and unanticipated aspects of network operation and management
- Be as much as possible independent of the implementation of particular hosts or gateways [5]

As a result, the main strengths of SNMP are its simplicity, interoperability, and low footprint on agents [6].

SNMP must also work effectively when the network is not fully operational. This reflects in the selection of a connectionless transport protocol (UDP), which allows management applications to exercise full control over the retransmission strategy.

Another design choice was to keep SNMP as independent as possible of other network services. This is one of the main reasons why, in SNMP version 3 (SNMPv3), security is self-contained and does not rely on other external security services such as key exchange or certification services.

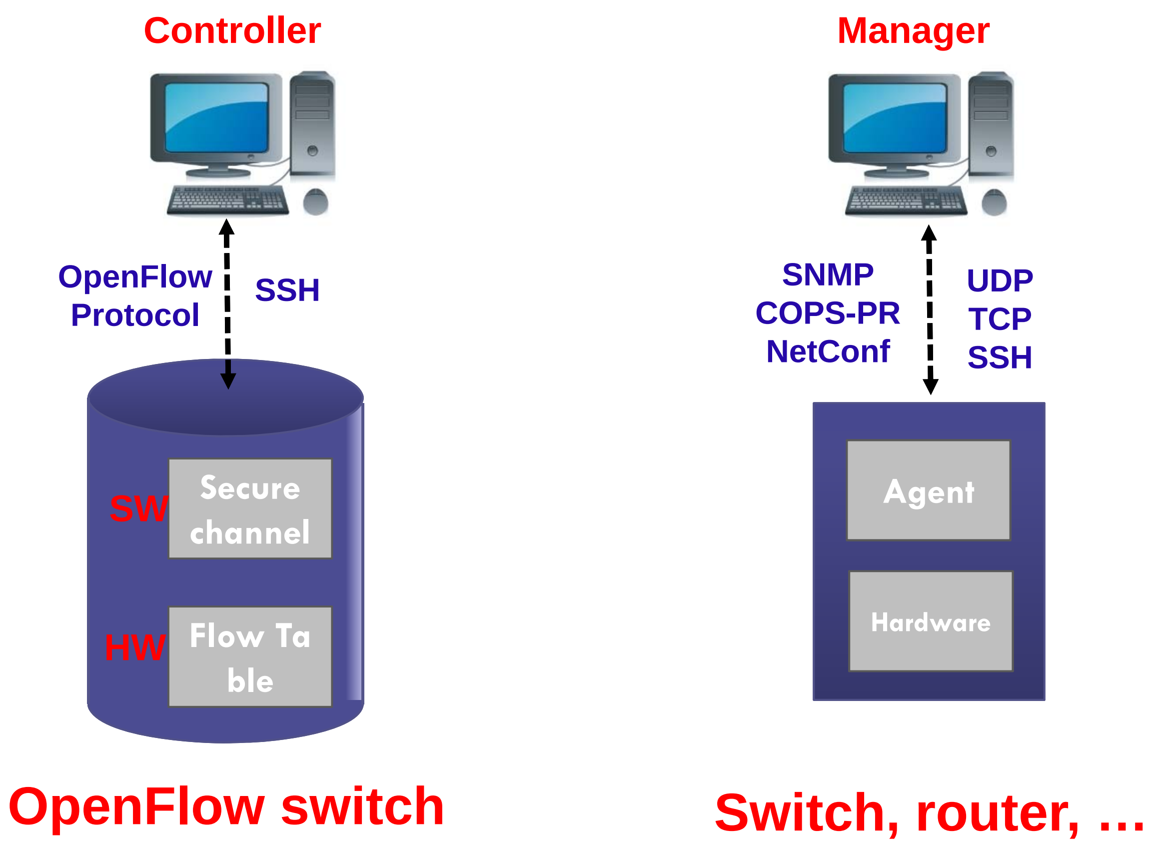
But the environment in which management operations take place has dramatically changed since SNMP was devised. Looking at today's network technologies and the actual usage patterns of SNMP, it is obvious that devices could perform more complex management operations at low cost. It is reasonable to expect that devices, especially high-end routers and switches, will become increasingly programmable, and that it will become possible to execute more control software directly on the devices.

Furthermore, as described by Wellens and Auerbach [7], SNMP need not use UDP. When network connectivity is lost, non-SNMP mechanisms are usually used to bring back connectivity before management operations can resume.

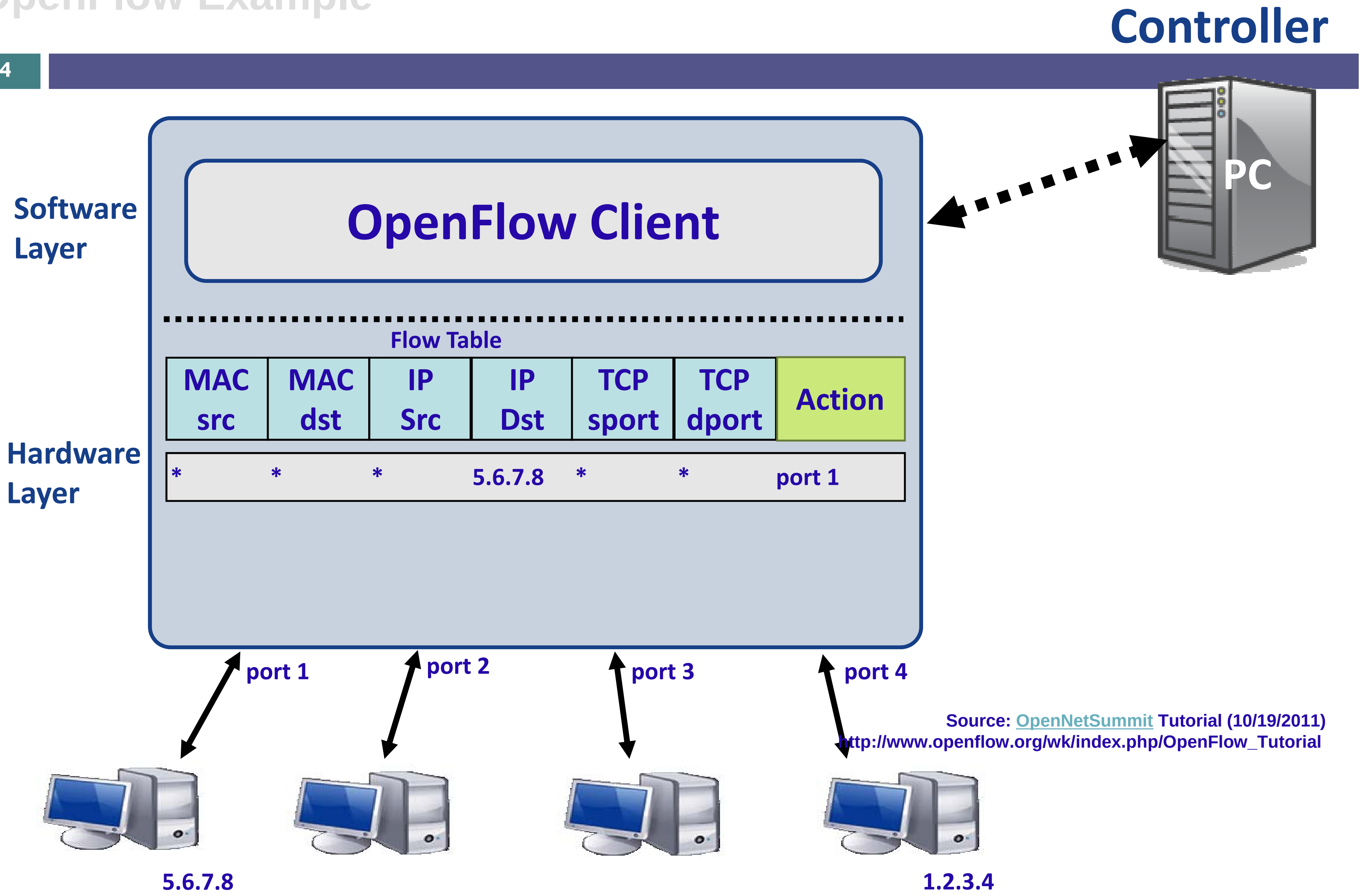
Finally, SNMP was standardized at a time

ComMag 2003

OpenFlow characteristics



OpenFlow Example



Source: OpenNetSummit Tutorial (10/19/2011)
http://www.openflow.org/wk/index.php/OpenFlow_Tutorial

SNMP – OpenFlow

- Both support configuration of (Forwarding) tables
  - Similar to SNMP's Interface and IP Forward MIB (inetCidrRouteTable)
  - Interface numbers are fixed, however
  - Therefore no support for "dynamic" interfaces
- OpenFlow 1.0 protocol is inflexible (no IPv6, ...)
- OpenFlow 1.1 protocol is more flexible
  - Similar to SNMP's VarBind idea
  - Better separation of protocol and data
- OpenFlow can mark several commands as "atomic"
  - Begin atomic / end atomic
  - Somehow comparable to SNMP's SET atomicity
- General agreement that SNMP is too low-level for configuration management

COPS-PR – OpenFlow

- COPS-PR Technology Comparable To SNMP
  - Objects Have Higher Granularity (Table Rows)
  - Single Operation To Add Or Delete Table Rows
  - Reliable Communication Between PDP And PEP (Because Of TCP)
  - Each PEP is Connected to Single PDP
- OpenFlow approach is quite similar to COPS-PR
- In 2002 IAB stopped COPS-PR

NetConf – OpenFlow

- NetConf has strong separation between protocol and configuration data
  - Standardization of configuration data is slow
  - Easy to extend in case of new configurations
- OpenFlow has tight integration
  - Easy to understand
  - Hard to extend
- NetConf has rich set of PDUs
  - Commit / rollback is possible
- Unclear what concepts of OpenFlow are better than NetConf

Configuration Protocols Conclusion

- OpenFlow is (yet another) configuration management protocol
- OpenFlow has many similarities to:
  - SNMP
  - COPS-PR
  - NetConf
- Granularity level of OpenFlow is:
  - Higher than SNMP
  - Same as COPS-PR
  - Lower than NetConf
- OpenFlow somehow mixes PDUs and Data
  - Easier to understand
  - Harder to extend
- Network management research community should use their expertise to improve OpenFlow design

Outline

- GENI Working Groups for Future Internet Mgmt
  - Control Framework
  - Experiment Workflow & Services
  - Instrumentation & Measurements
  - Operation, Management, Integration & Security (OMIS)
    - GMOC GENI Meta Operation Center

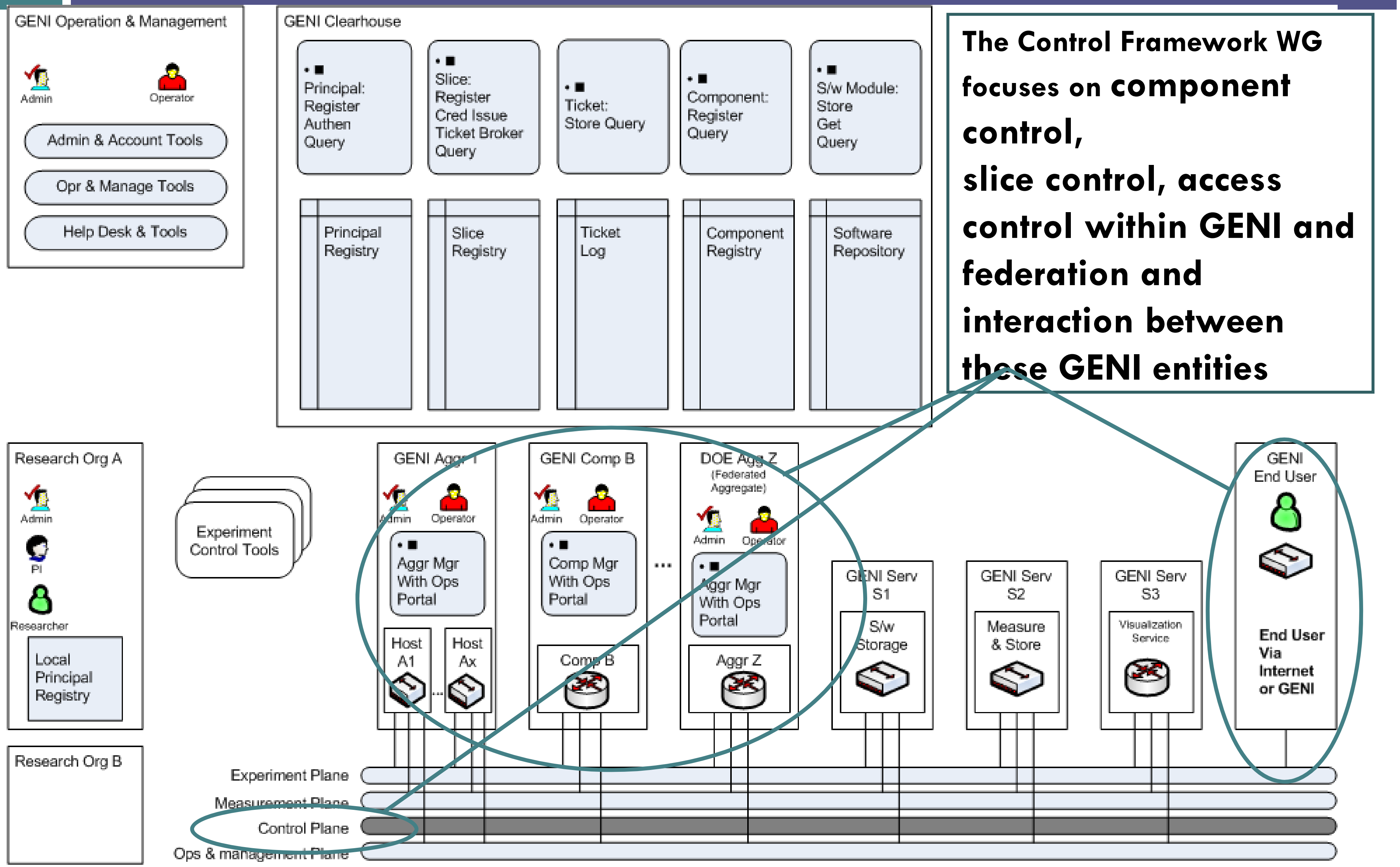
GENI Working Groups

- Control Framework WG
  - Logically stitching GENI components and user-level services into a coherent system
  - Design of how resources are described and allocated and how users are identified and authorized
- Experiment Workflow and Services WG
  - Tools and mechanisms a researcher uses to design and perform experiments using GENI
  - Includes all user interfaces for researchers, as well as data collection and archiving
- Instrumentation & Measurements WG
  - GIMS - GENI Instrumentation and Measurement Service
  - GENI researchers require extensive and reliable instrumentation and measurement capabilities to gather, analyze, present and archive Measurement Data
    - To conduct useful and repeatable experiments
- Operations, Management, Integration and Security (OMIS) WG
  - Designing, deploying, and overseeing the GENI infrastructure
  - Operation Framework

Control Framework

- GENI control framework defines:
  - Interfaces between all entities
  - Message types including basic protocols and required functions
  - Message flows necessary to realize key experiment scenarios
- GENI control framework includes the entities and the Control Plane for transporting messages between these entities
  - component control
  - slice control
  - access control within GENI
  - federation
  - key enablers such as identification, authentication and authorization

GENI Architecture - Control Framework

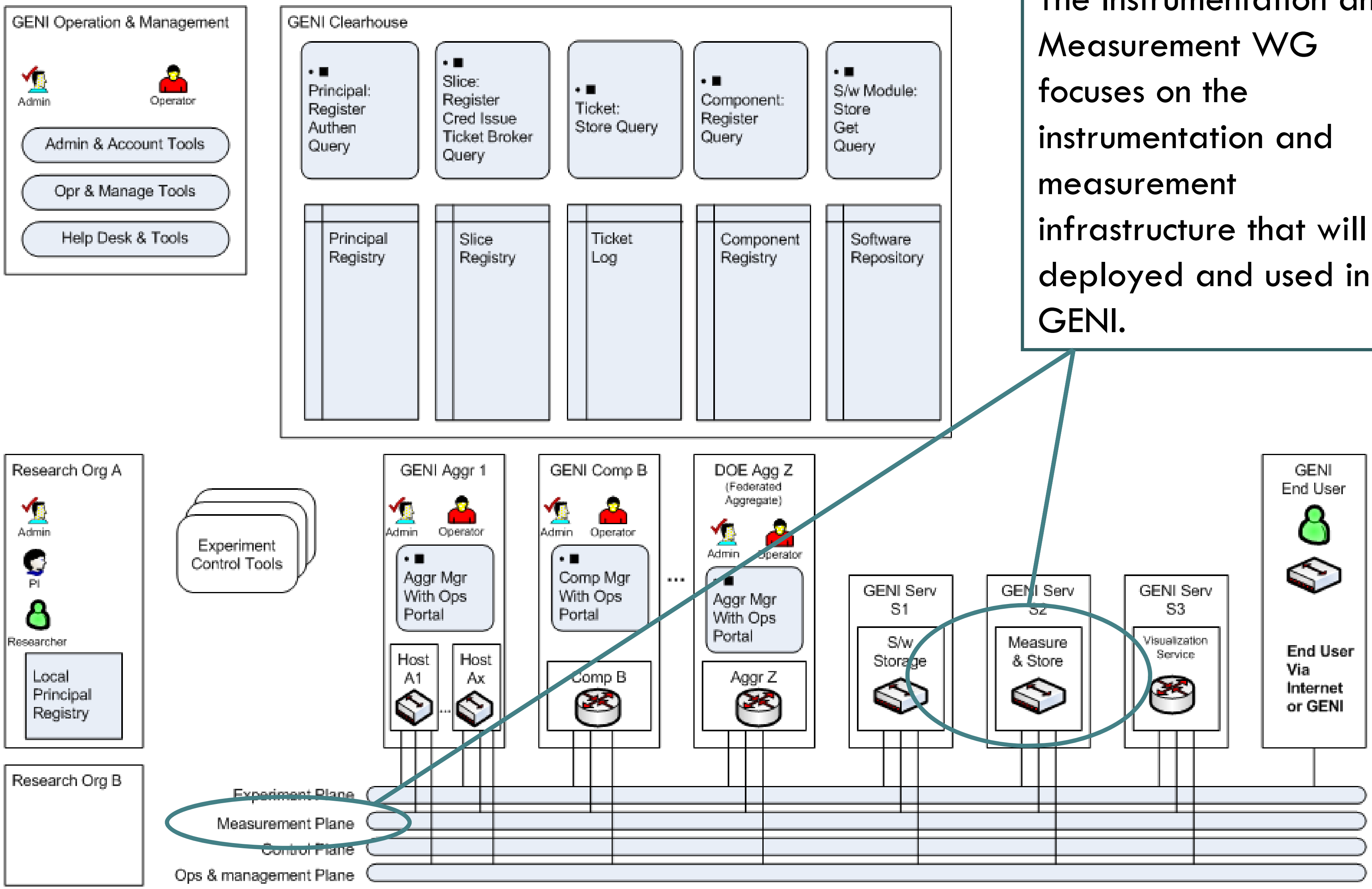


The Control Framework WG focuses on component control, slice control, access control within GENI and federation and interaction between these GENI entities

Instrumentations & Measurements

- Discuss, develop and build consensus around the architectural framework for the instrumentation and measurement infrastructure that will be deployed and used in GENI
- Create an architecture for measurement that enables GENI goals to be achieved
- Facilitate dialog and coordination between teams focused on I&M
- Identify key challenges in I&M that could otherwise inhibit the infrastructure
- Solicit feedback from users
- Deploy basic instrumentation and measurement capabilities
- Services
  - Measurement Orchestration (MO)
  - Measurement Point (MP)
  - Measurement Collection (MC)
  - Measurement Analysis and Presentation (MAP)
  - Measurement Data Archive (MDA)

Relationship to GENI Architecture



The Instrumentation and Measurement WG focuses on the instrumentation and measurement infrastructure that will be deployed and used in GENI.

GIMS – Protocols & Communication

- Researcher via Experiment Control service (tools), including MO(Measurement Orchestration) service, manages the setup and running of I&M services
- Protocols for researcher/experiment control tools to access APIs:
  - Xml-rpc
  - web services (SOAP, WSDL)
  - APIs for setting up and running I&M services
  - APIs for MP (Measurement Point) services
  - APIs for MC (Measurement Collection) services
  - APIs for MAP (Measurement Analysis and Presentation)services
  - APIs for MDA (Measurement for Data Archiving) service
- All traffic is carried in the GENI Control Plane

GIMS Traffic Flow

- Option 1:
  - Carry all MD (Measurement Data) traffic flows using a dedicated measurement VLAN
- Option 2:
  - Carry all MD traffic flows using the same IP network that supports the Control Plane.
- Option 3:
  - Carry most MD traffic flows using the same IP network that supports the Control Plane, but for high-rate MD traffic flows, define a dedicated measurement VLAN for the slice/experiment

Detailed Outline for OMIS

- Operation, Management, Integration & Security (OMIS)
  - GMOC GENI Meta Operation Center
    - Why Meta-Operation?
    - Objective
    - Architecture
    - Operational Data Set
      - Topology
      - Operational Status
      - Administrative Status
      - Utilization Measurements
      - Specialized Data
    - Data Acquisition & Sharing
    - Communication & Coordination
    - Operations
    - Use Case
      - Notification
      - Emergency Shutdown Functions

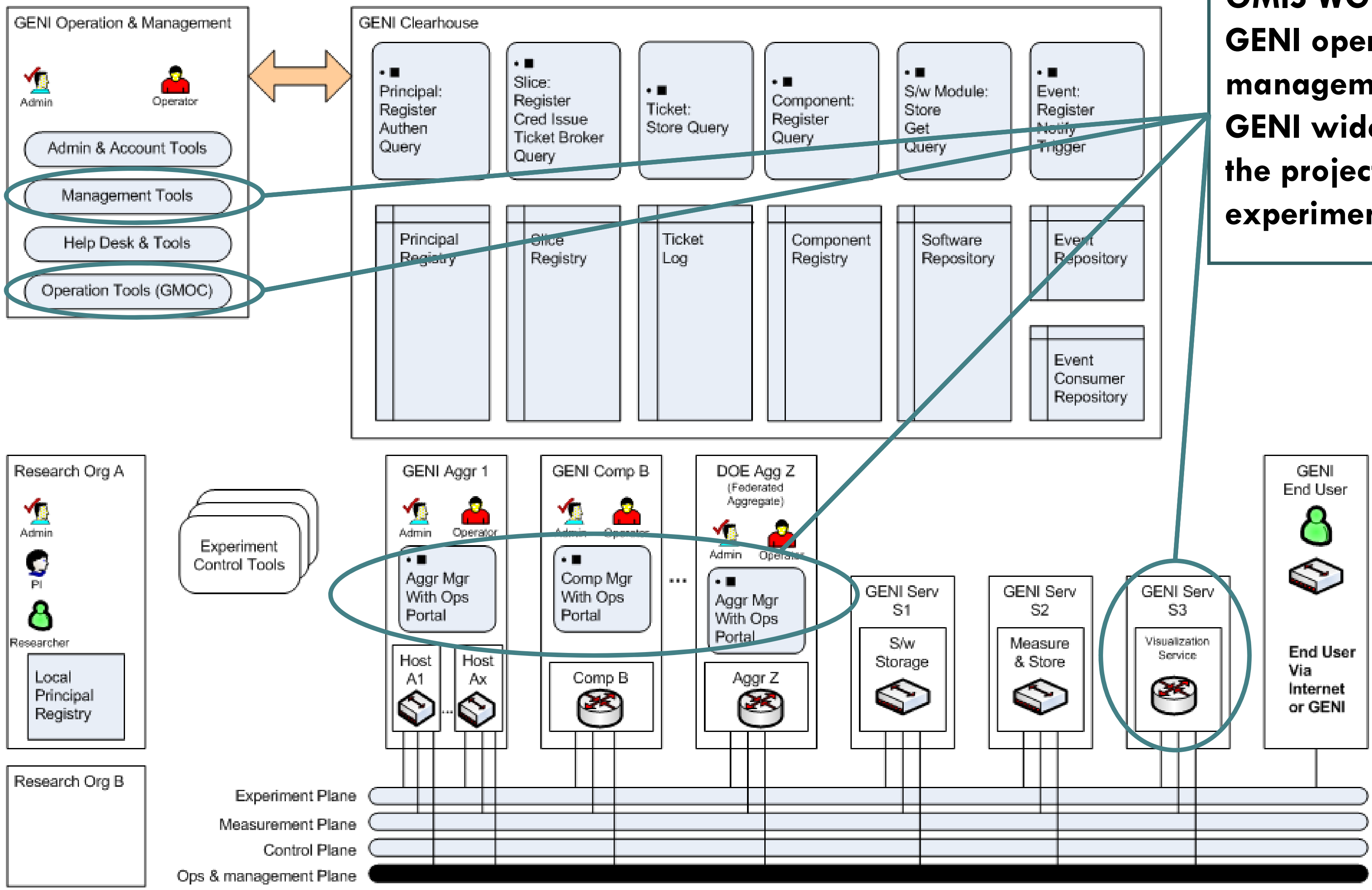
OMIS

- Operation
  - GMOC (GENI Meta-Operation Center)
- Management
  - Meta-Management System for GENI
- Integration
  - Overlap & Interfaces with other WGs
- Security
  - Policies, Authorization & Authentication

Overlaps with other WG

- Control Framework WG
  - common interface for operations
  - Security
    - lower levels of GENI & higher level should be consistent
- Experiment Workflow and Services WG
  - Operation & Management Tools
  - Services Usage
- Instrumentation & Measurements WG
  - Data Acquisition
  - Measurements for performance and management

Relationship to GENI Architecture



OMIS WG focuses on GENI operations, management and GENI wide view of the projects and experiments

Question and Discussion