# A Polymorphic Network Architecture based on Autonomous Domains

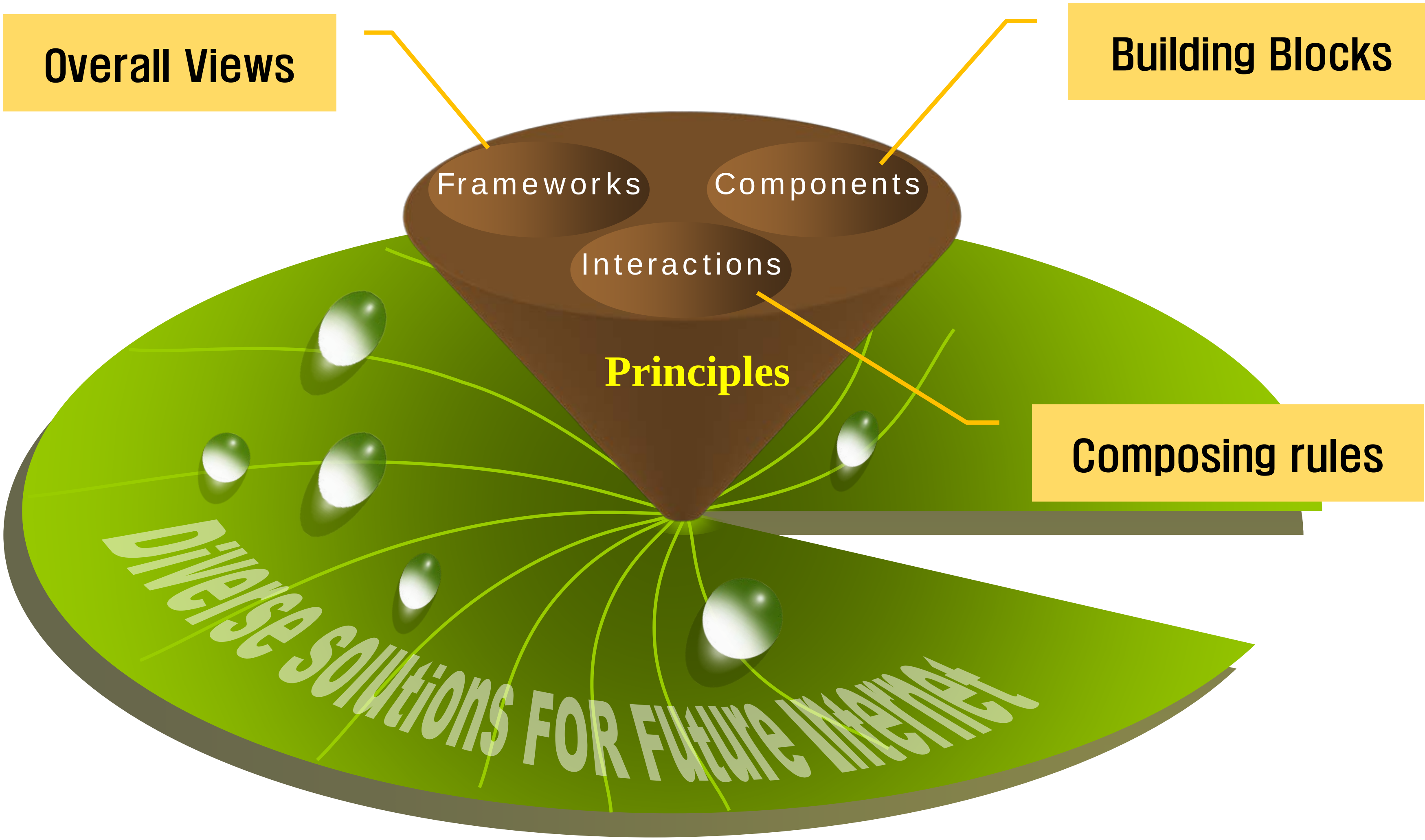
## DIANA

**Domain-Insulated Autonomous Network Architecture**

Electronics and Telecommunications Research Institute

Woojik Chun

# Target of the Work



# Scope of the Work

## Objectives/Requirements

- What we want – Functional, Structural & Quality

## Principles

- What to Choose – Selection criteria, Priority, Norms

## Framework / Model

- How to Design – Style and Rules

## Components / Interactions

- How to Build – Building Blocks and Glue logic

# Requirements & Principles

1 Requirement & Principles

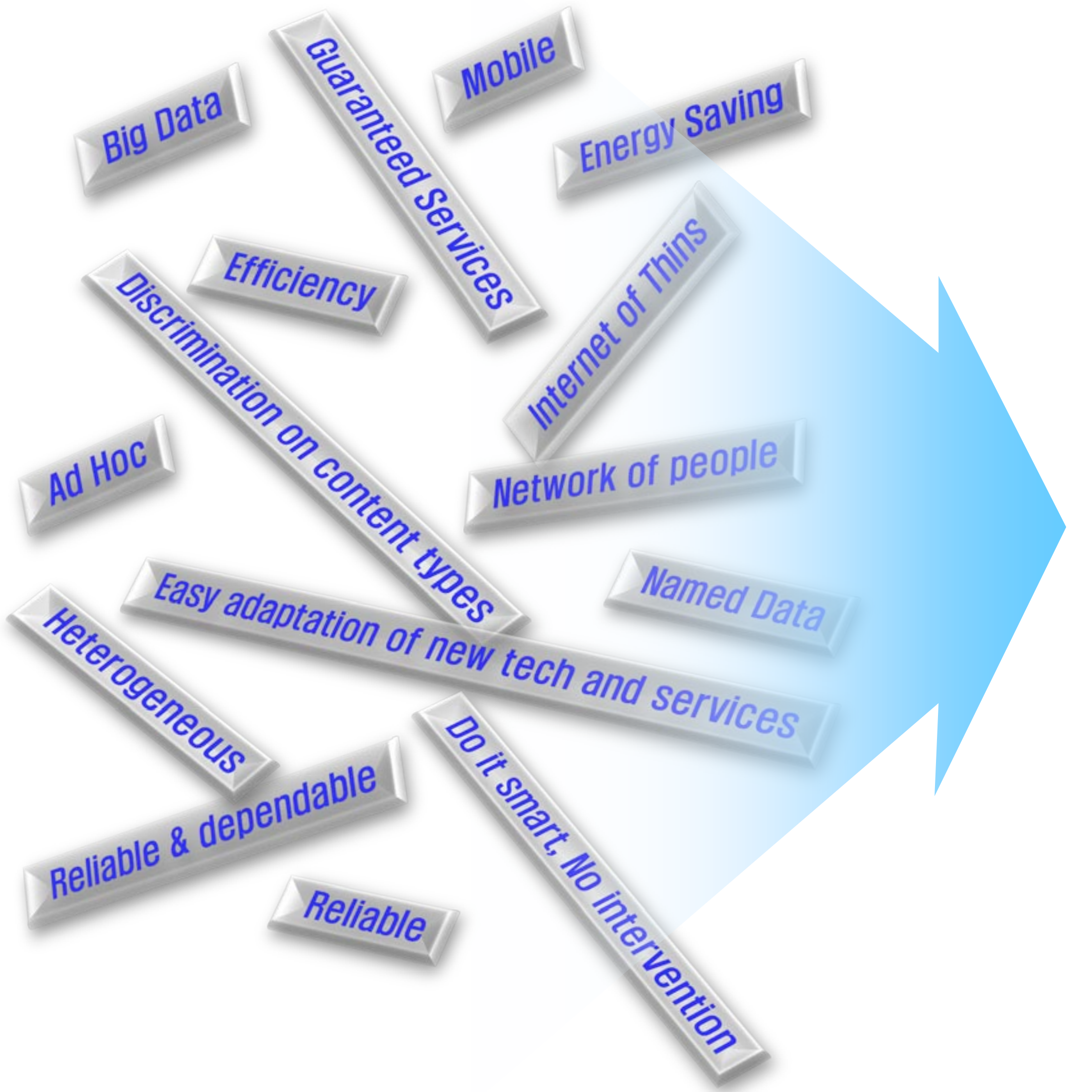
- 6 S Requirements
- Principles

2 Framework & Model

3 Components & Interactions

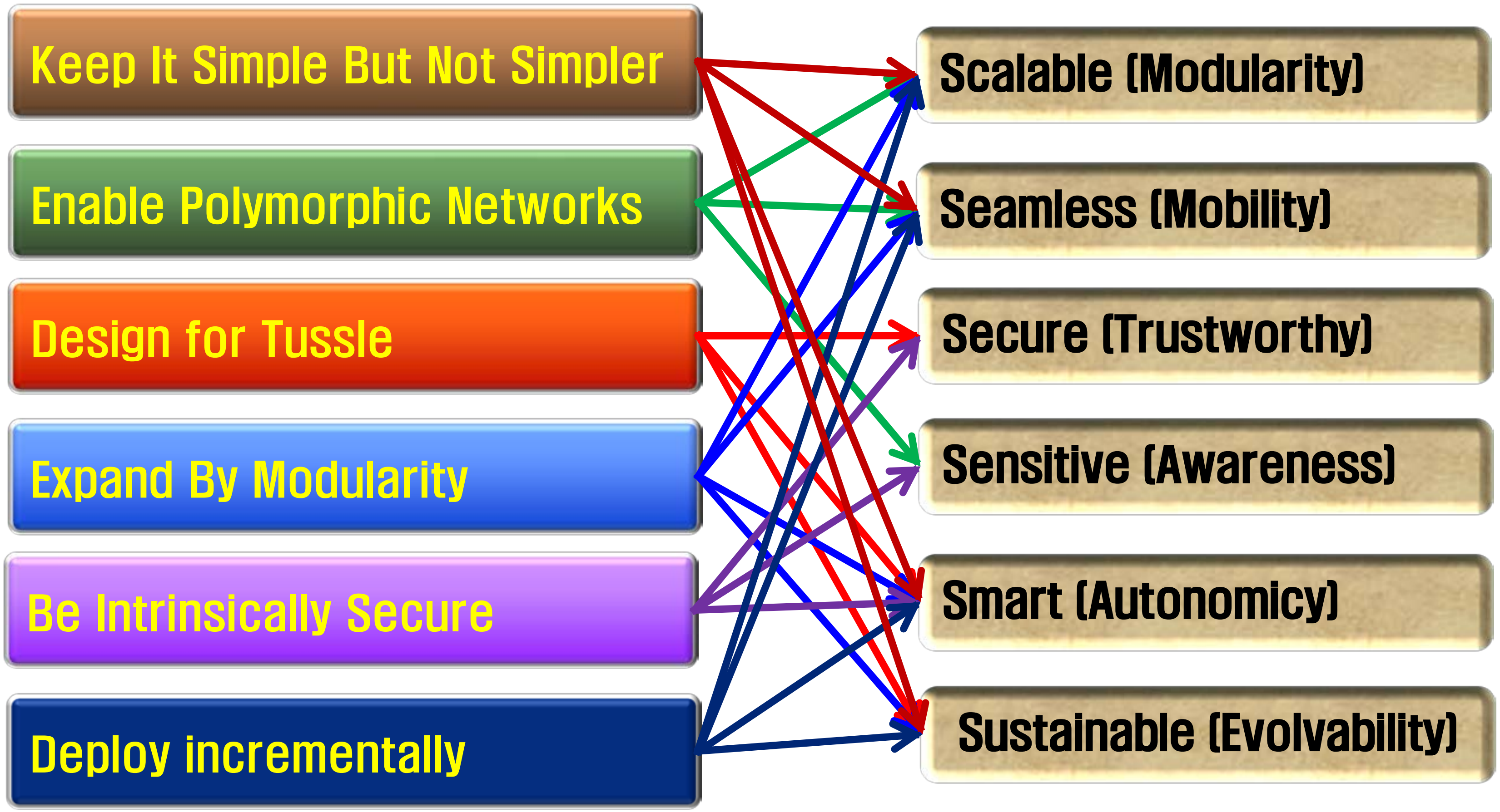
4 Putting Together & Applying

# 6S Requirements




- Scalable (Modularity)
- Seamless (Mobility)
- Secure (Trustworthy)
- Sensitive (Awareness)
- Smart (Autonomicy)
- Sustainable (Evolvability)

# Principles for Requirements



# Principles: Features Derived

| | |
|---|---|
| 1. Simple | • ID for Device, User, Contents, Services, and Things<br>• ID based Bus Abstraction ➔ Narrow Waist |
| 2. Polymorphic | • Autonomy for heterogeneous networks<br>• Dynamic logical network creation |
| 3. Tussle | • Union by Federation instead of unification<br>• Simple rules for Federation |
| 4. Modularity | • Horizontal layering as well as vertical layering<br>• Recursive definition of Network Composite |
| 5. Secure | • Trust level and required security<br>• Self-certifying ID |
| 6. Deployment | • Migration Plan<br>• Incremental deployment |

# Framework & Model

1. Requirement & Principles
2. Framework & Model
   - Polymorphic Networks
   - ID based Networking
   - ID Lookup Servers (ILS)
3. Components & Interactions
4. Putting Together & Applying

# Why Architectural Framework?

Ref: Architecting for Innovation, 2011 T.Koponen, S. Shenker, etc

- Motivation
  - Removing barriers to innovation
    - Architectural modularity
  - Inter-domain routing
    - Universal glue that makes end-to-end connectivity in global scope
  - Security
    - Built-in security features (as well as trustworthy framework)
  - Additional
    - Meta-negotiation between two hosts (possibly communicating entities)
    - Bootstrap interface that can be called at initiation phase

Ref: The Future Networking, 2011 S. Shenker

- Why Framework?
  - When first getting systems to work ➔ Focus on mastering complexity
  - When making system easy to use ➔ Focus on extracting simplicity
- Future of networking lies in finding *right abstractions*
  - The era of “a new protocol per problem” is over

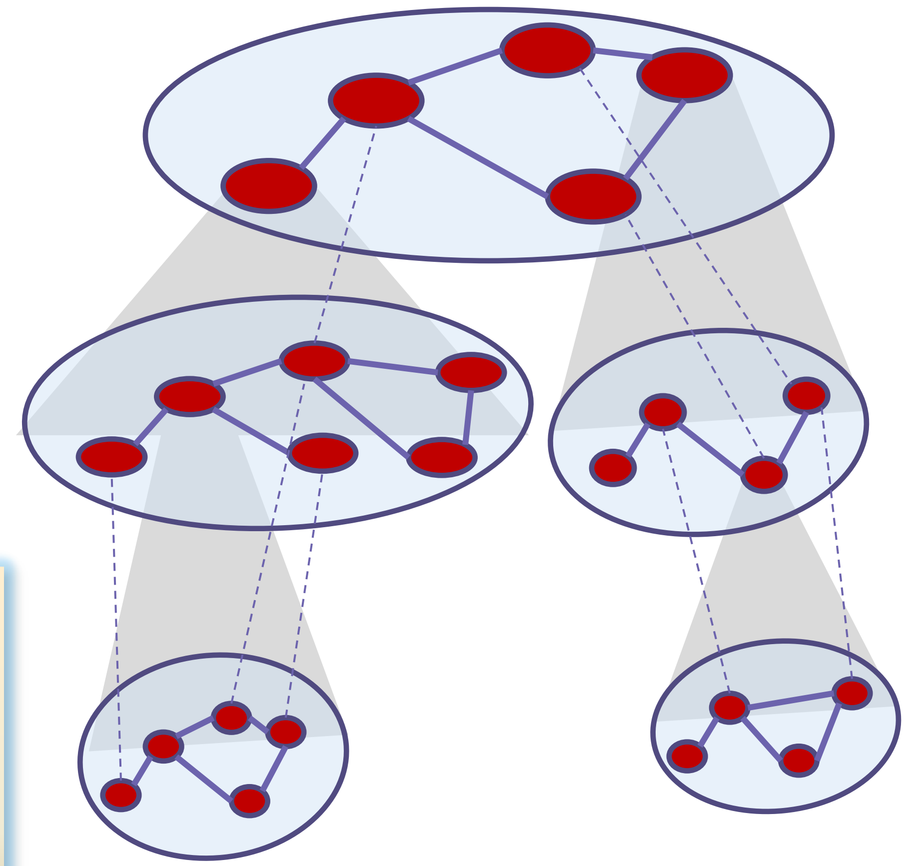
# Methods – Abstraction & Recursion

## Abstraction

- ❑ a process by which *higher* concepts are derived from the usage and classification of concepts
- ❑ while hiding the internal details

## Recursion

- ❑ a method calls itself
  - ■ To solve a simpler version of the problem
  - ■ To repeat the same principles
- ❑ Self-Similarity
  - ■ Looks like Fractals



# Polymorphic Networks

- What is Polymorphic Network
  - A network where one could implement, and deploy its new network protocols or cooperation schemes without disturbing other working protocols
- Why Polymorphic ?
  - Various research projects, scientists and “people” will propose new ideas ➔ heterogeneous architectures
  - Enable different cooperation paradigms in parallel
  - Enable easy deployment of new application deployment
- How ?
  - Virtualization, Overlay, Abstraction, Mapping …
  - Without raising routing and addressing to the application
    - As Peer to Peer networks, overlay networks, VPN, Spontaneous networks ….

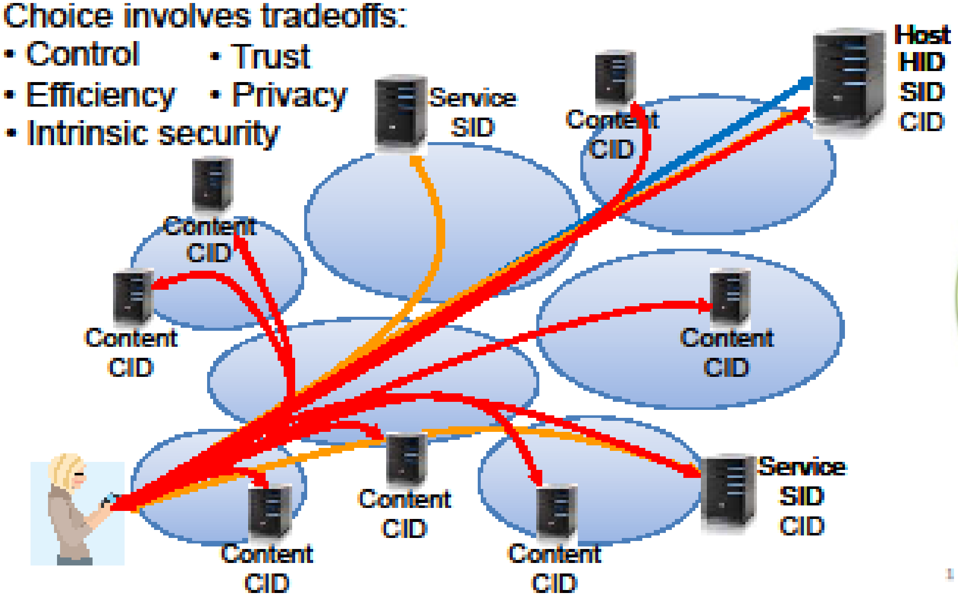
# Polymorphic Networks – Examples

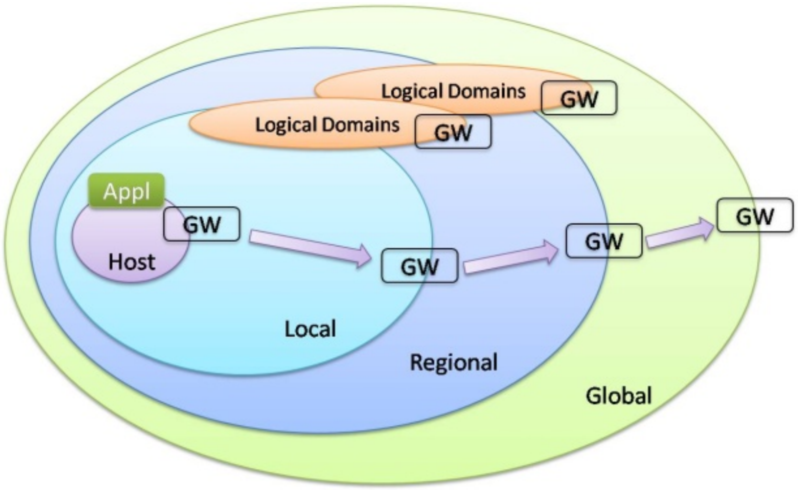
## Multiple Principal Types

- x-Centric
  - HID, CID, SID, UID …
- Different trust levels
  - guarantees depending on principal type

## Network Composites

- Horizontal Layers
  - Local, Regional, Global
- Logical networks
  - Virtual networks
  - Spontaneous networks



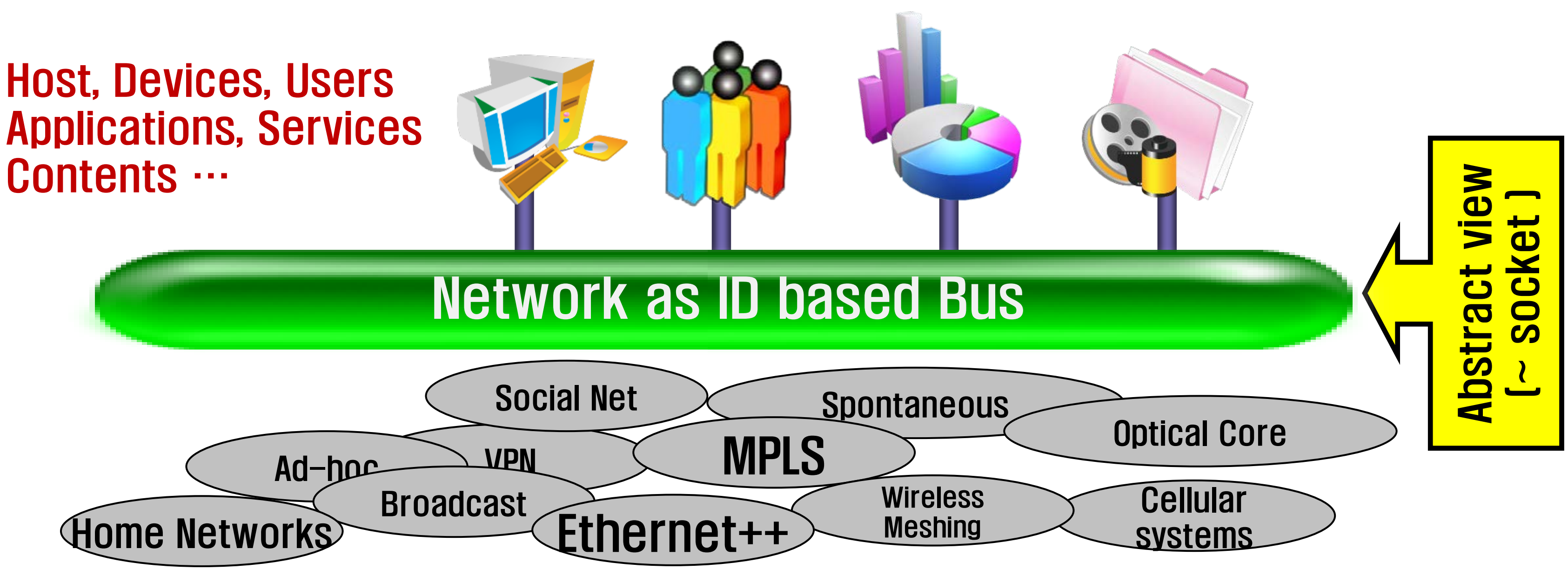



# Polymorphic Networks – Issues

- How to define individual networks ?
  - Define dissimilar networks as independent domains
  - Provide minimal rules to allow communication among dissimilar networks or logical groups (network composites)
  - Internal details of a domain are hidden from outside
  - Domains can be unit of self-management (Self-*)
  - Domain may associated with other domains
    - in vertical manner (parent-child relation)
    - in horizontal manner (peer relation)
  - Domain has one or more gateways (interfaces)
- Challenging topics
  - Life cycle management for domains
  - Federation of domains

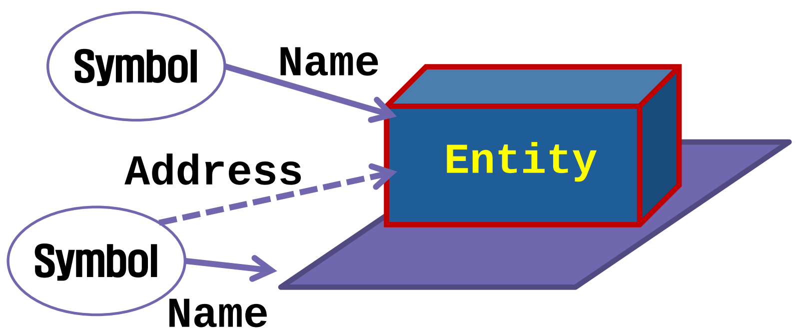
# ID based Networking

- ID to all communicating Entities
  - hosts, services, and contents are plug in network bus with globally unique ID
- Location-independent ID
  - ID must be bound to location (early or late binging)
- provide well-defined interfaces
  - register, publish, present, associate, send/receive

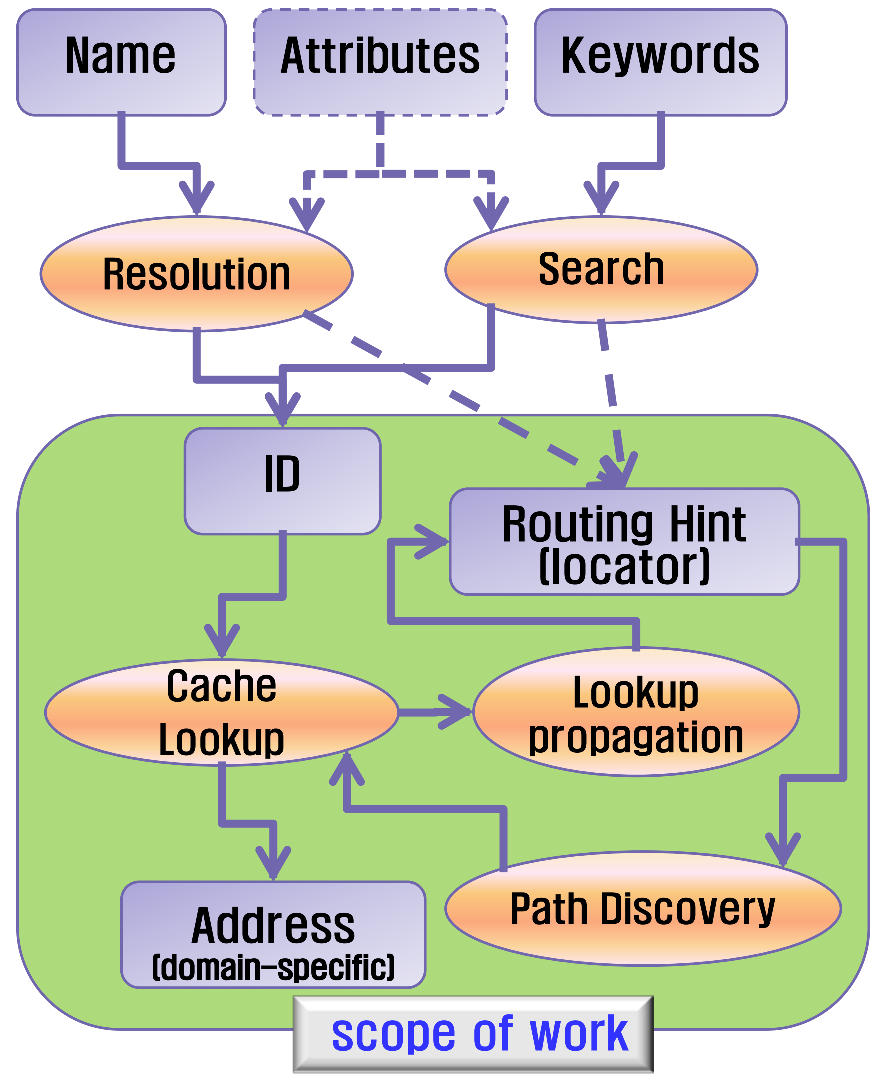


# ID based Networking : Definitions

- Definitions by Assignments
  - Name → what
  - Address → where




- Definitions by Roles
  - ID → scope
    - Unique in a given scope
  - Locator → space
    - relative in a given space
    - Vector, metric, topological space



# ID based Networking – Issues

- ID
  - Location-independent ID
  - Self-Certifying
  - Same syntax for all entities (devices, users, contents, services …)

- ID based socket
  - API for application developers

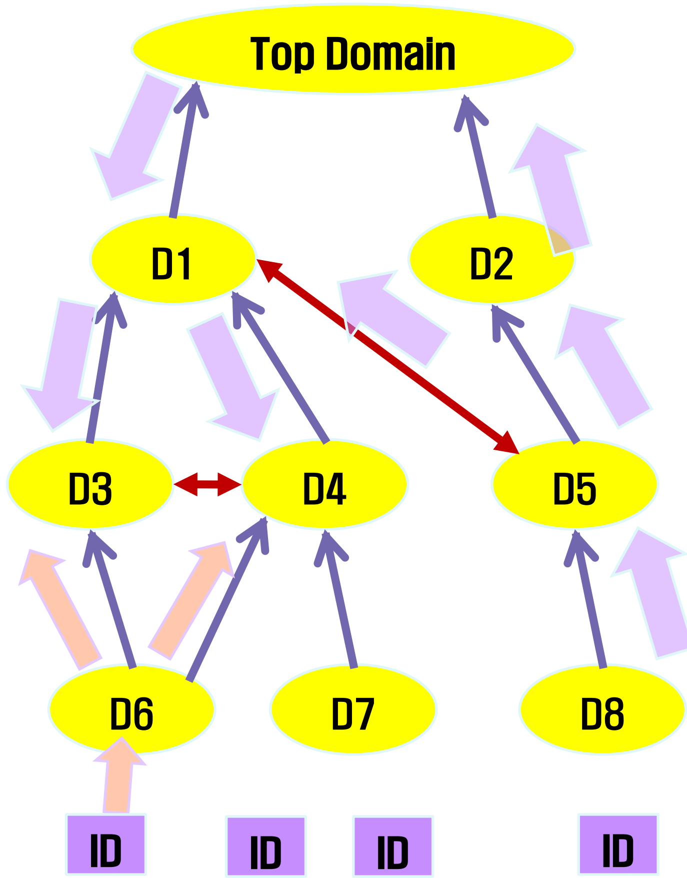
- Challenging topics
  - ID based Forwarding (because of ID's location-independency)
  - Proactive vs. Reactive routing
  - Forwarding Entry Explosion (because of Non-aggregatable)
  - Routing Scalability (because of no. of IDs in the network graph)

# ID Lookup Servers

- ID-LOC separation
  - ID has no clue on its location
  - To be accessed, ID should be bound to LOC
- 2-Step Binding
  - Registration (Enrollment)
    - Communication Entity(CE) must register its ID to one or more domains
    - each domain has "ID registry function" → (ID, locator) binding
    - ID becomes "valid" by registration
  - Attachment (Presence)
    - CE actually appears at the certain domain
    - the registered domain may not be the present domain
    - at the point of presence, an address of the presenting domain is assigned (either early or late binding)

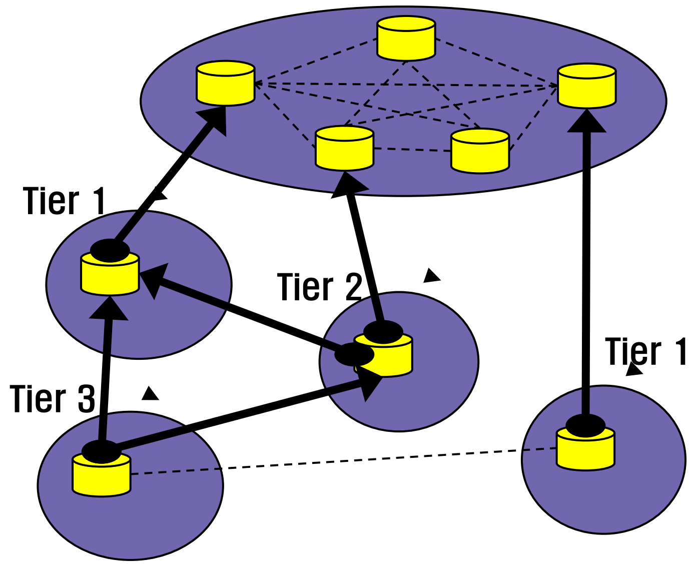
# ID Lookup Servers : Registration and Lookup

- ID registration
  - first registered at a domain
  - registered ID may be propagated toward its ancestor
    - till a domain in between home and top domain
    - downward lookup propagation from top domain finishes at the first domain that know the ID
    - need balance between registration and lookup propagation
- ID lookup
  - default to parent (optionally peers)
    - not always same as data forwarding path

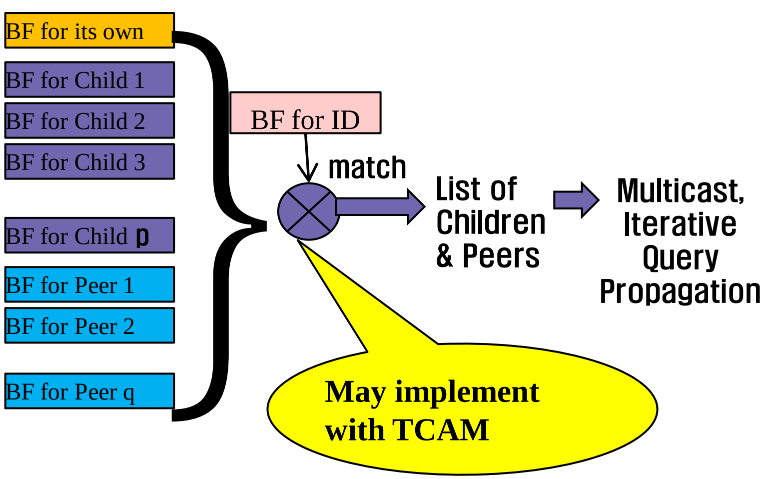


# ID Lookup Servers : Issues

- ID Lookup Server (ILS)
  - Each domain is associated with an ILS
  - Distributed, Hierarchical ILS structure
    - Top-Level-Tier (fully peered)
    - Intermediate Tiers (Parental-child, Peer relations)
    - Leaf (only child relation)
  - ILS has relation with others
    - Parent-Child, Peer relation




- Challenging topics
  - Optimization on query propagation
    - Use Bloom Filter



# Components & Interactions

1 Requirement & Principles

2 Framework & Model

3 **Components & Interactions**

- Domains
- ID based Routing
- Self-Certifying

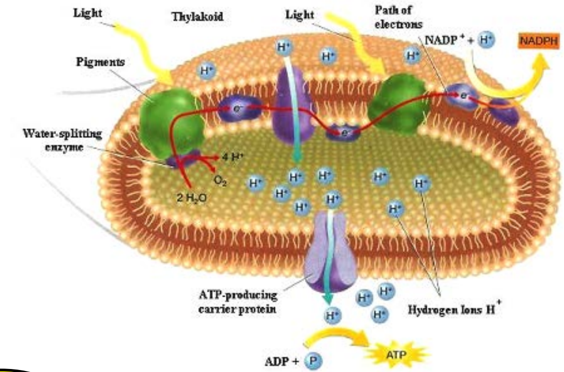
4 Putting Together & Applying

# Domains

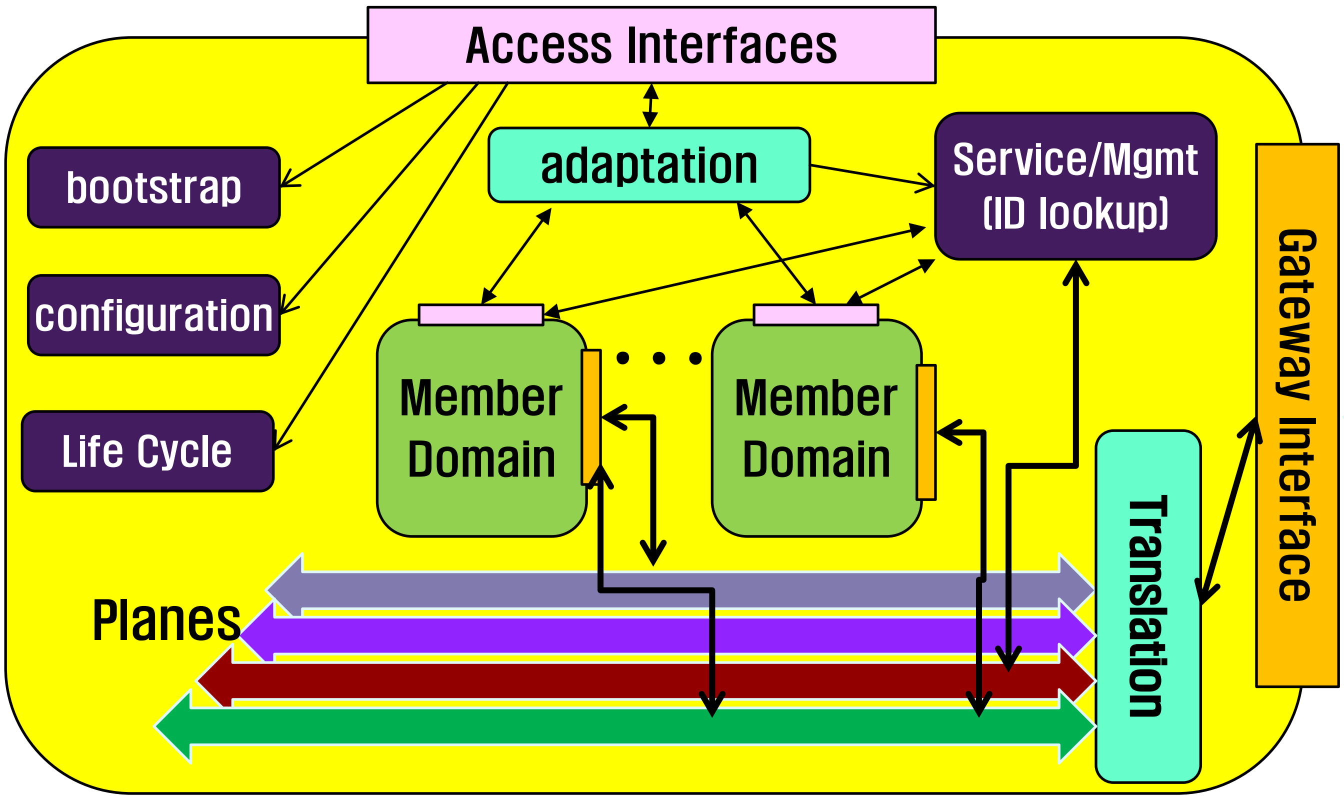
- What is Domains ?
  - Basic Building Blocks for composing the whole network
  - Abstraction for a piece of the network (network composite)
    - Separate administration, Different media, Local network, Logical networks, Spontaneous networks, Group of services …
    - Similar concepts: Principal, Turf, Compartment, Context, DIF …
  - Unit of Autonomy
    - Internal operations are hidden from outside of the domain
    - Domain-specific routing
    - May have self-* capability
  - Unit of Insulation
    - Accessed via well-defined interfaces
  - Unit of Federation/Composition
    - May be defined in recursive manner
- Horizontal layers of the network
  - Build Fortress on Prairie

# Domains: Basic Building Blocks

- Abstract View of a Domain
  - Recursive Definition
  - Adaptation / Translation features
  - Self-manageable




Origin of Cells
In Sea of Data

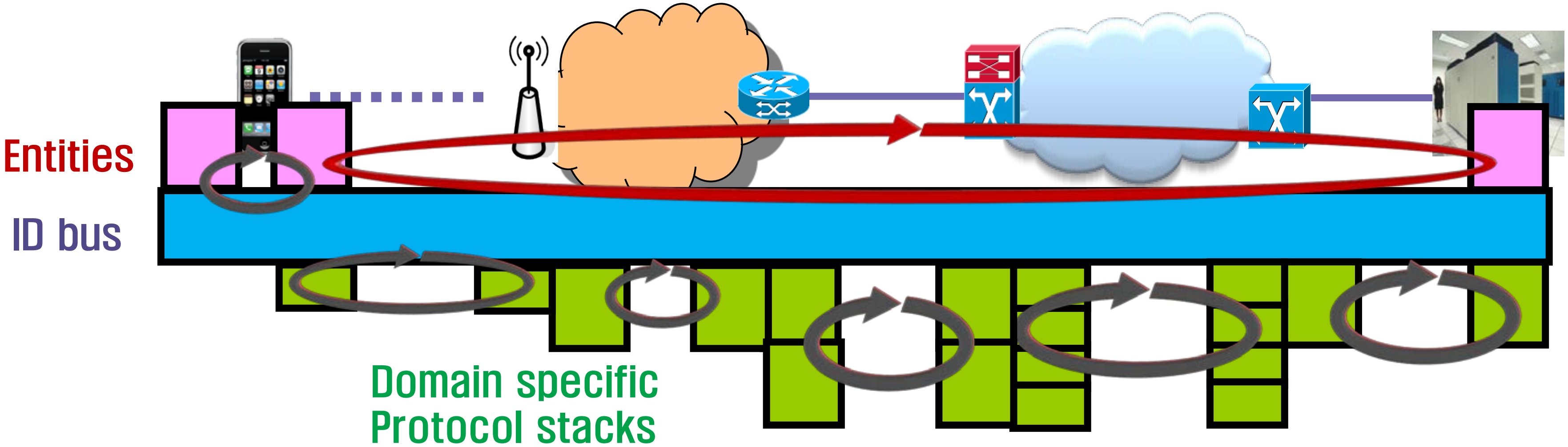




Inference protocol
Management protocol
Control protocol
Transport protocol

# Domain : Justification

- Each domain provides functions, only where being required
  - Consider why middle boxes violate layer principles
- Examples
  - IPC within a host
    - only process ID
  - Access to a server within the same LAN
    - need Host ID + MAC address
  - Access to a host at the site with local IP address
    - simple fixed routing table (no routing protocol)
  - Access to a server through internet
    - forward to default gateway (NAT) and use IP and routing



# ID based inter-domain Routing

- Design Goals
  - To provide mechanism for inter-domain routing
    - Intra-domain routing is domain-specific
  - Define minimal set of rules
  - Support mobility, multi-homing, trustworthy, late binding …
- Distinguishing Features
  - Reactive routing with built-in entries
    - Forwarding entries are setup when actually required
    - Some frequently used entries may be built in advance
  - Forwarding Cache with timeout
    - Entries are setup by explicit "path discovery"
    - Unused entries are removed in timeout basis
  - Path Discovery with routing hints
    - The path for an ID is setup using the location as a routing hint
    - Utilizing ILS for finding (ID, LOC) binding

# ID based Routing : Basic Concepts

## ❑ Packet Format (PDU)

- Invariant part: no modification throughout inter-domain
- Variant part: changed depending on domain specific mechanism

| Type | Domain Address Source | Domain Address Destination | Domain Options | Type | EID Source | EID Destination | E2E Options | Payload |
|---|---|---|---|---|---|---|---|---|
| Variant part | | | | Invariant part | | | | |

## ❑ Routing Hints

- Location (and policy) information bound to the ID
- Obtained by query to ID lookup servers (ILS)
- Normally, Fully Qualified Domain ID (FQDI)

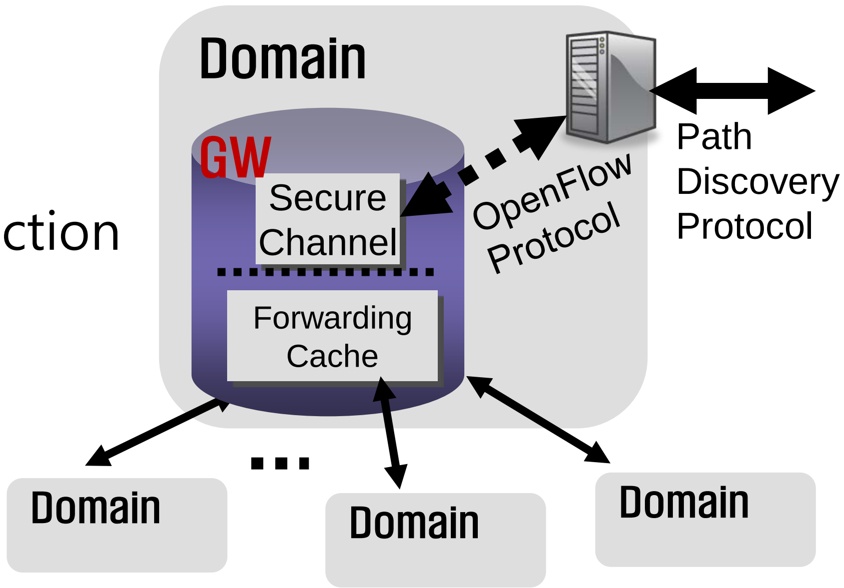
  "Registered Domain @ …. @ Top-tier Domain"
- Used in "path-discovery"

# ID based Routing : Basic Concepts

- Forwarding Cache (FC)
  - Indexed by ID (exact match)
  - Setup by path-discovery
  - Domain specific forwarding

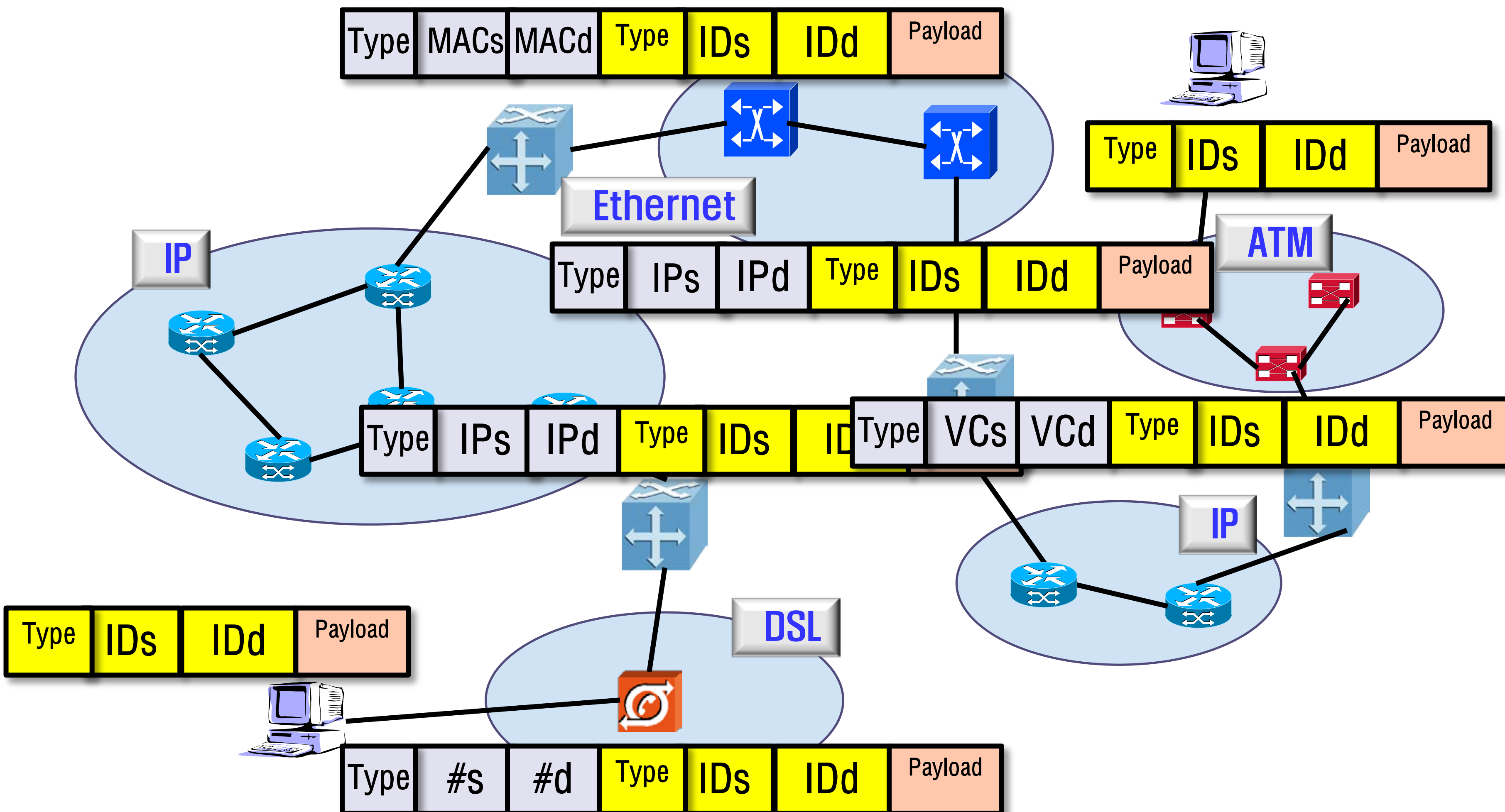
| ID | Dev | Type | Domain Address |
|---|---|---|---|
| B1 | GW | Eth | 00:01:02:03:04:05 |
| C1 | GW | IP | 129.168.55.22 |
| D1 | GW | UDP | 129.168.100.3:2485 |
| E1 | GW | TCP | TCP connection ID |
| *A1* | *ID* | *direct* | – |

- Path Discovery
  - triggered by the first packet when no entry in FC
  - may be initiated by control function
  - On successful path discovery
    - Setup entries on the GW along the path
    - May utilize "OpenFlow"



# ID based Routing : a simple example

- Assume entries at the GW along the path have been setup



# ID based Routing : Issues

- ID exact match (flat ID)
  - No longer LPM(longest prefix match) ➔ high performance
  - No aggregation ➔ scalability ?
    - Keep only relevant entries in FC
    - Per-interface forwarding cache (split interface)
- Location-independency
  - (ID, LOC) binding requires ID lookup propagation
    - Delay at the first packet
    - Frequently-used entries may be built in FC in advance
  - Late-binding and multi-homing is possible
    - Multi-homing : ID with multiple locations
    - Identical copies (servers) : same ID to multiple contents (servers)
- Policy based "path-discovery"
  - May enforce policy while path discovery
  - Resource reservation can be done while path discovery

# Self-Certifying

- Intrinsic Trustworthy Framework
  - Self-certifying ID for every communication entity and domain
  - Insulated Domain
    - All End-point ID (EID) should be verified
    - All communication must pass the well-defined gateway
  - Trusted Domain
    - Trust-level between domains
- EID verification
  - At the first-hop domain
  - Only verified ID can be installed in FC

- Domain verification
  - If domain A trust domain B in a certain trust-level
    - Then performs appropriate functions (forward or block)
  - Initiate security functions depending on trust-level

# Components & Interactions

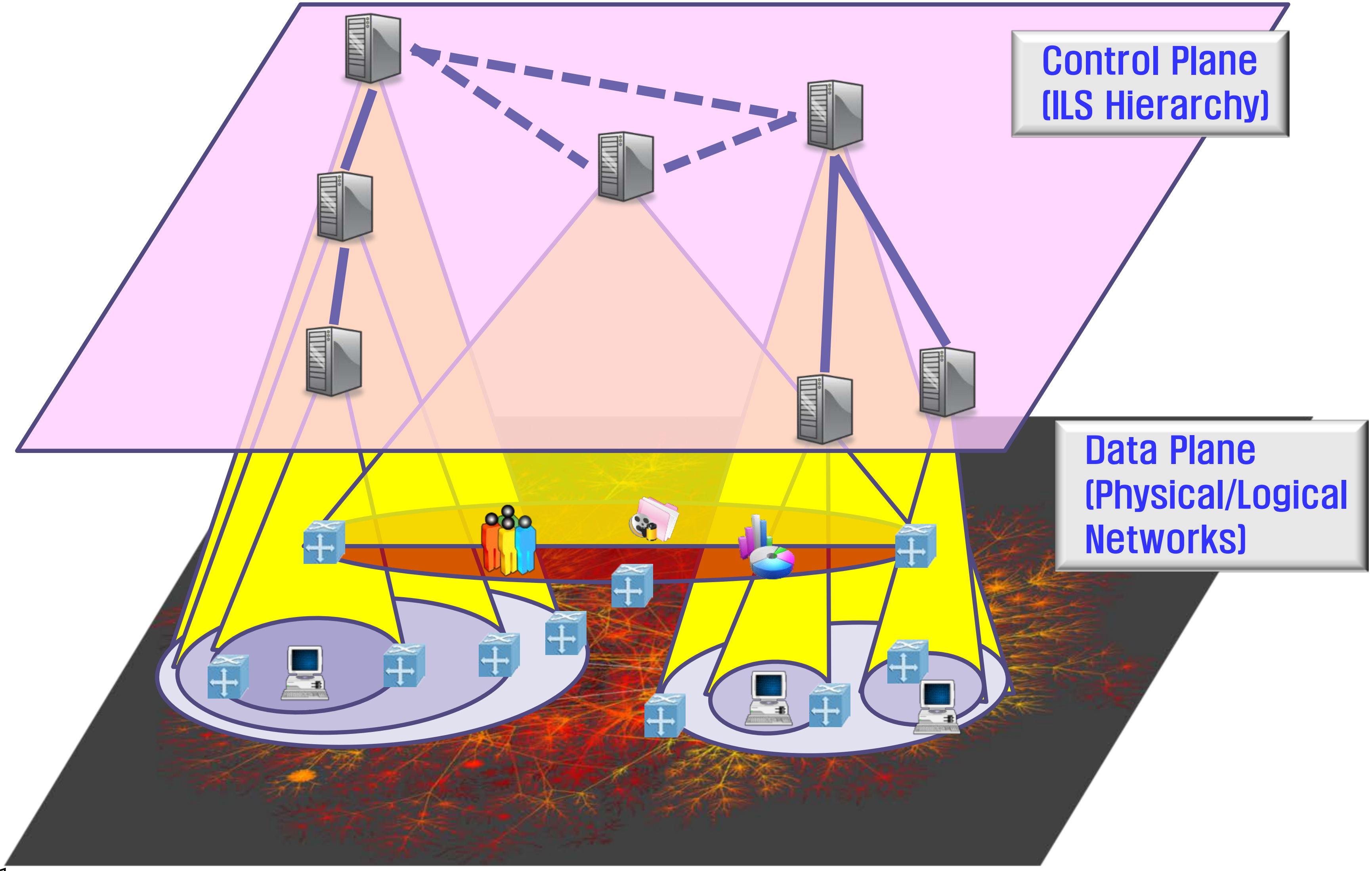
1 Requirement & Principles

2 Framework & Model

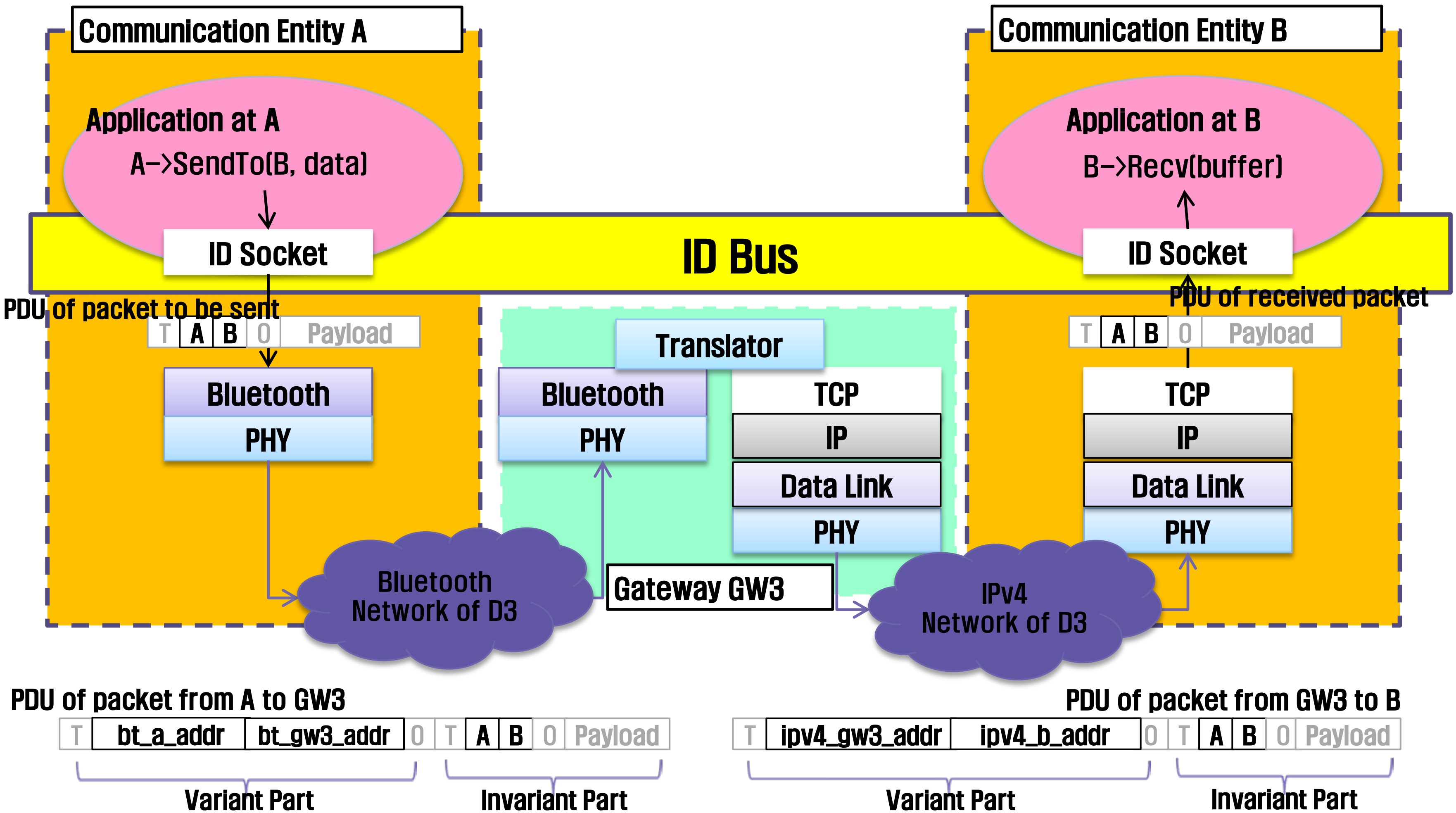
3 Components & Interactions

4 **Putting Together & Applying**

# Putting Together: Data & Control Plane

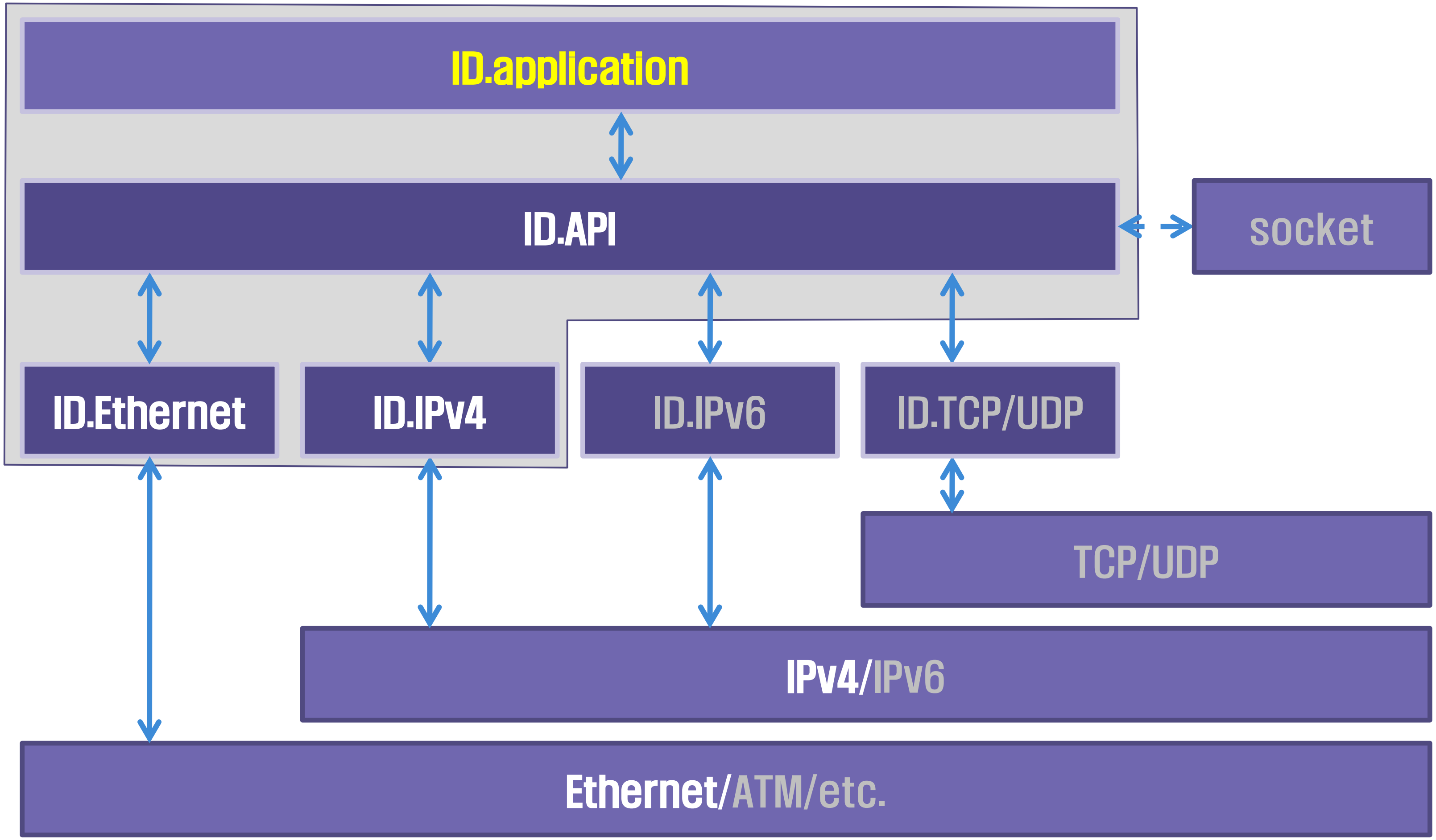


# Putting Together: ID Socket



# Putting Together : Prototyping

- Implement on Linux Kernel
- Two-level abstraction (ID & IP/Ethernet)



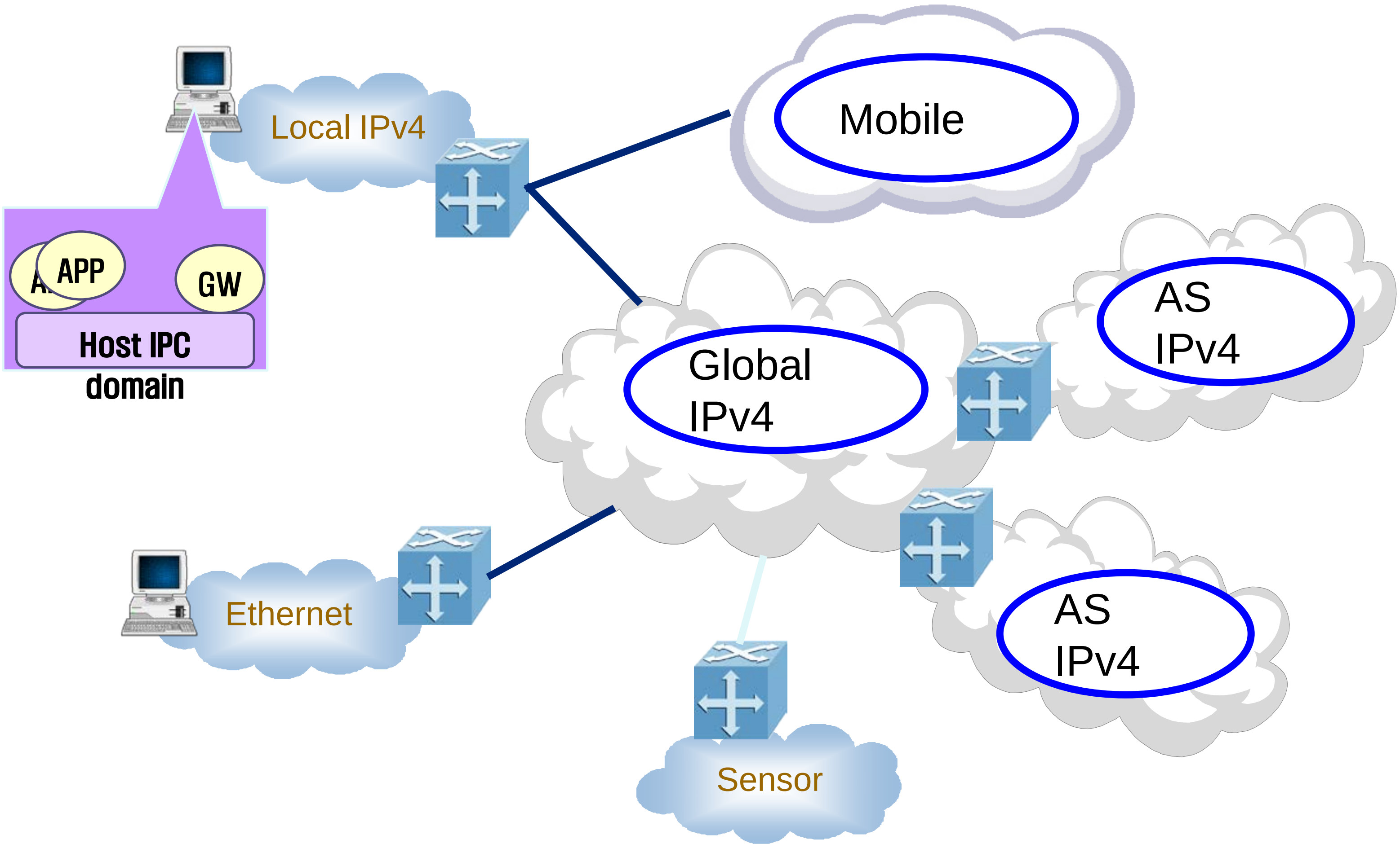
# Putting Together: Applying

- Multicast, Anycast, Concast
  - Defined as logical domains (with attributes)
  - Define root of the forwarding tree as GW of the domain
  - Join/Leave → register/de-register from the domain
  - Forwarding Tree setup → path discovery
- Mobility
  - Present → Location update at the ID Lookup Server
  - Move → exception → triggering "path discovery"
- Named Data Network
  - Named Data → assign ID (Content ID)
  - Use the "ID based routing"
  - Identical Copies are handled by the same way as multi-homing
- VPN, P2P, Spontaneous Networks
  - Dynamic creation of domains and membership management

# Putting Together: Applying

- Active entities and Passive entities
  - Active entities : ID + processing (Protocols)
  - Passive entities : ID without processing
    - forms a domain with GW
    - Agent(without ID) at the GW
- Software Define Network
  - Must concentrate on control/management functions
  - Borrow software development techniques
    - Object oriented (Class & Instance)
    - Recursive definition
    - Abstraction and virtualization
    - Late binding
  - Apply the principles to network composite as well as networking functions
    - Network composite == Domain

# Putting Together: incremental Deployment



# Major Contributions

- Autonomous Domains
  - Map polymorphic networks onto independent domains
  - Keep trust zones based on self-certifying ID
  - Accommodate dissimilar architectures by simple rules
- Hierarchical structuring of Domains
  - Allow recursive definition of network composites
  - Eliminate routing scalability issue
- Reactive routing and Path-Discovery
  - Reduces the burden of forwarding entry explosion problem
  - Allows forwarding table split by per-interface forwarding cache
  - Supports mobility and multi-homing (identical-copies)
- Incremental Deployment
  - Allows a number of domains on the current Internet
  - Accepts new networks without interference to the existing ones

# Concluding Remarks

- What FI has to do for new services?
  - Build shared infra with richer capabilities
  - Show incentives to adaptors of new concepts
  - Suggest graceful migration plans from the existing Networks
    - Not "backward compatibility" but "adaptability"

- New Era of Networks
  - Internet must be based on Science
  - Two major techniques
    - Abstraction
    - Recursion

- Network Composites
  - Network → Lego blocks
  - Union instead of unification
  - Simple rules for composition
    - Invariant glue logics