

# Gea: The Harmony.

**Information Infrastructures:  
Production and Distribution.**

[https://youtu.be/FjKiT\\_4cn2g](https://youtu.be/FjKiT_4cn2g)



Developed countries are evolving towards something usually called the “*Information Society*” but, What does it mean?

Here an attempt to present a model in a short and graphical way: a representation of how different components are integrated into a coherent whole.

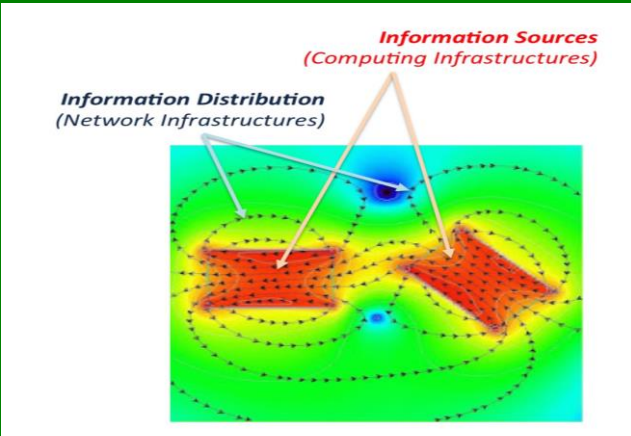
Which is the best use for all these information tools?

This will demand an entirely new theoretical framework for social systems design.

Practical results can't be seen in a lifetime period.

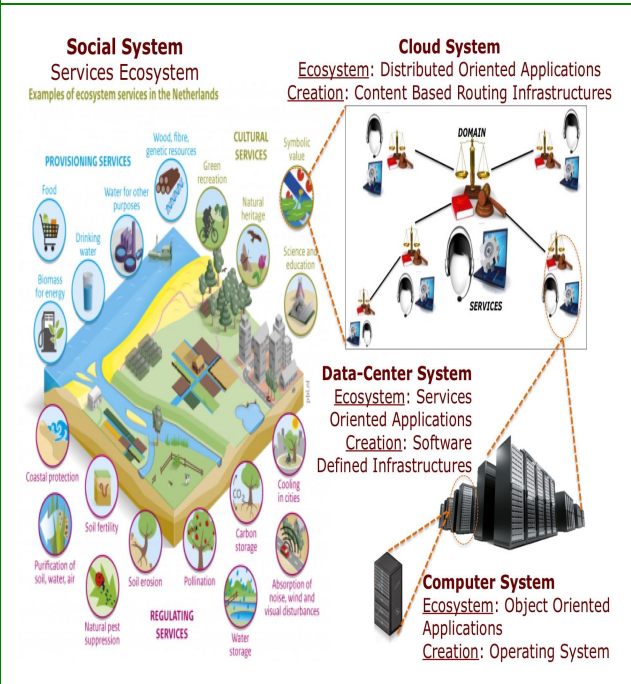


INFORMATION INFRASTRUCTURES SYNTHESIS: PRINCIPLES OF HARMONY



Principles of harmony rule the stratification process of infrastructures. The Russian Doll effect in two types of information distributed systems that cooperates to create a complex organism:

- **Information Sources:** computing infrastructures; different types of programmable compact units: a) computers, b) data-centers with associated repeaters and networks, c) federations of data-centers or clouds.
- **Information Distribution:** network infrastructures.



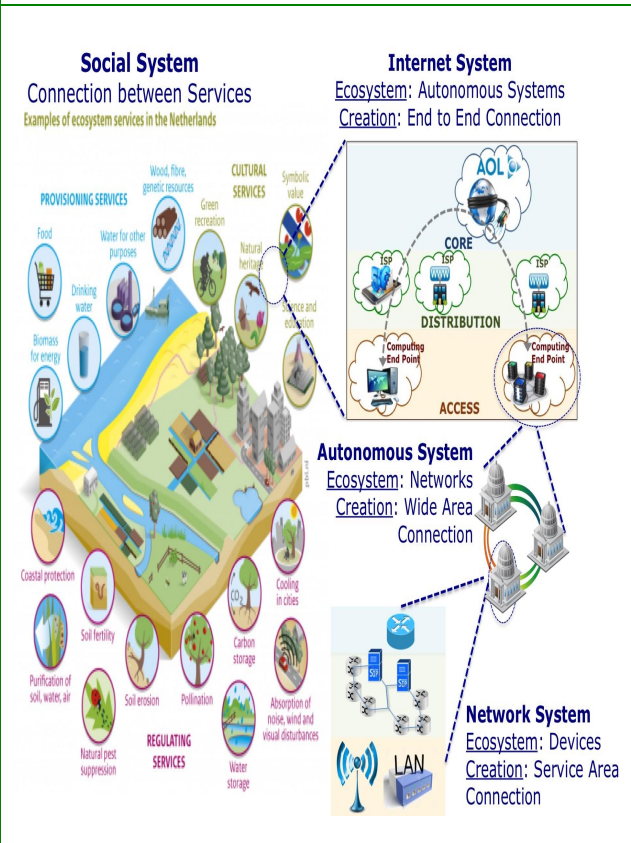
INFORMATION SOURCES: COMPUTING INFRASTRUCTURES

**SW Creation Basic Needs:** Programs are written in files while executed in processes.

**SW Articulation:** Operating System Scheme of Roles massively articulate process and file values.

**SW Types, three aggregation levels:**

- **Cloud System:** DOA = Distributed Oriented Architecture, cooperation schemes between data-centers to create distributed applications (ie. Google, Amazon, LinkedIn, etc.).
- **Data-Center System:** SOA = Services Oriented Architecture, cooperation schemes between computers inside a data-center dynamically allocated according to traffic demands to create cluster applications (ie. Microservices).
- **Computer System:** OOA = Object Oriented Architecture, cooperation schemes between processes inside a computer to create monolithic applications.



INFORMATION DISTRIBUTION: NETWORK INFRASTRUCTURES

**Connectivity Creation Basic Needs:** circuits for real time, datagrams for volume.

**Connectivity Articulation:** Network Scheme of Access-Distribution-Core Roles massively articulate datagrams and circuits.

**Connectivity Types, three aggregation levels:**

- **Internet System:** E2EC = End to End Connection, connection between computing end points (information sources) through a network of Autonomous Systems (AS).
- **Autonomous System:** WAC = Wide Area Connection, connection between social entities through a network of service areas, in other words, connectivity across an autonomous system that plays a role within the internet system (ie.: computing end points, ISP=internet service provider, etc.).
- **Network System:** SAC = Service Area Connection, connection between devices, in other words, across a service area that plays a role within the autonomous system (ie.: SDH optical transport, MPLS traffic handling, IP networks).



### Social System

#### Services Ecosystem

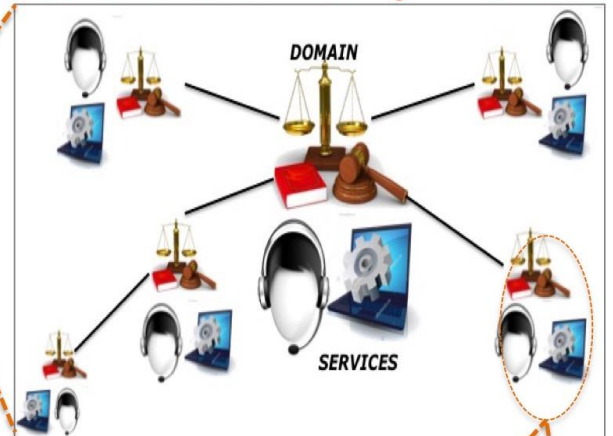
Examples of ecosystem services in the Netherlands



### Cloud System

Ecosystem: Distributed Oriented Applications

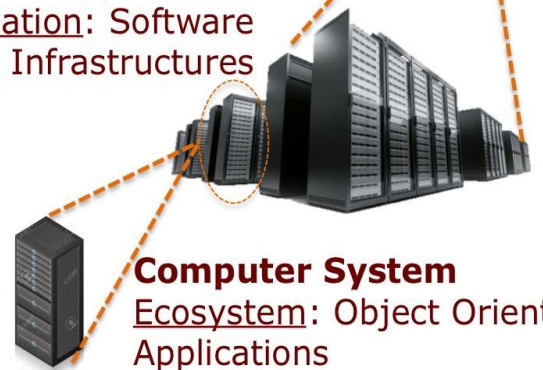
Creation: Content Based Routing Infrastructures



### Data-Center System

Ecosystem: Services Oriented Applications

Creation: Software Defined Infrastructures



### Computer System

Ecosystem: Object Oriented Applications

Creation: Operating System



# Computing Infrastructures.

## Information Sources

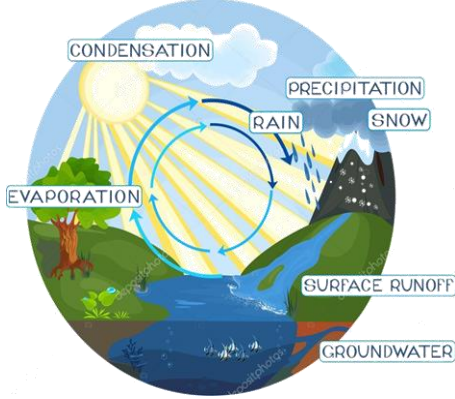


### COMPUTING INFRASTRUCTURES: IDEAL GOALS

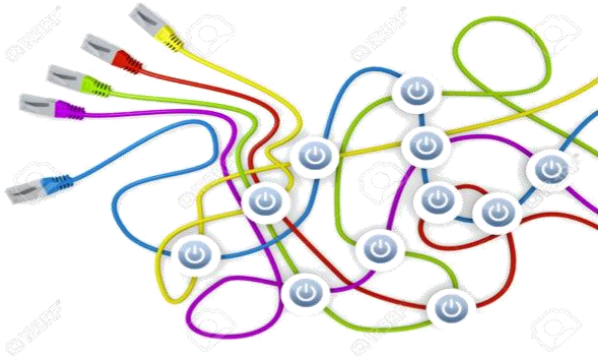
#### Services Creation

##### Organic Logic

Production: Organically Integrated working environments following well known system models



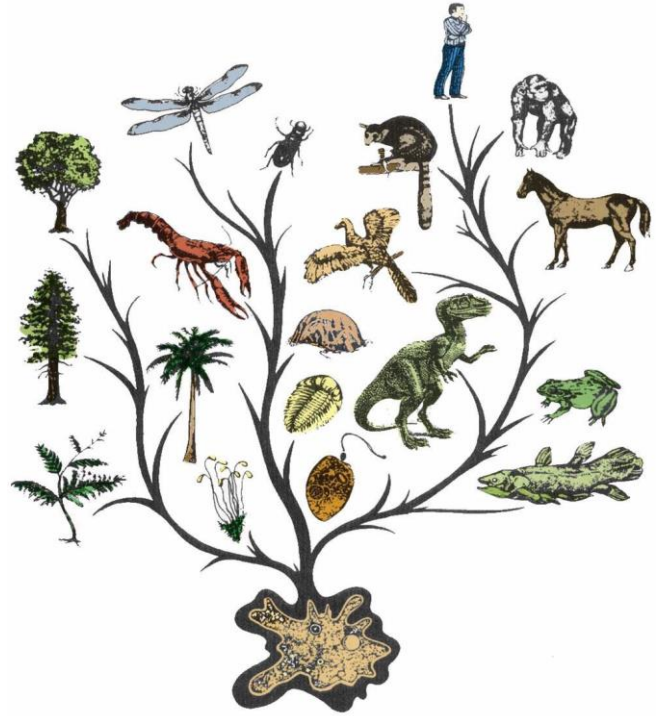
Distribution: clone & RTP connectivity



#### Services Ecosystem

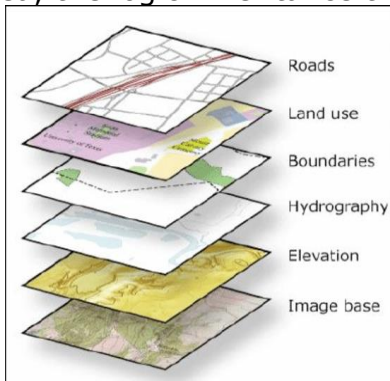
##### Single Common Root Framework

Entire tree of social systems in constant evolution with a coherent versioning control.



##### Content Based Routing Data

Production: Multilayer data deposition on each area, the logic inheritance tree.



Distribution: Content Based Routing across areas



##### Social Systems Modelling

Binary Tree Ordering:  
Private Articulation & Public Operation



**Development Infrastructures:**  
Computing  
Information Sources



**Private: Articulation**  
Economic  
Legal & Accounting  
Systems



**Public: Operation**  
Economic  
Production & Distribution  
Systems



**Cloud Type**



**SW Creation: Distributed Oriented Architecture**

Scalability Run-Times based on compartmenting in logical zones.



**SW Ecosystem**

Tree of Logical Domains: cloned desktop of cluster applications across domains.

**Articulation:**  
SW Creation Needs



**Process:** Service



**File:** Repository



**Signal:** RTP Network

**Operation:**  
Organic Distributed System



**Run:** Software Defined Function Cloning



**I/O:** Software Defined Content Based Routing



**Connect:** SD-WAN, Software Define Wide Area Network



**Data-Center Type**



**SW Creation: Services Oriented Architecture**  
Microservices Run-Times based on Continuous Delivery & Continuous Integration



**SW Ecosystem**  
Desktop of Cluster Applications

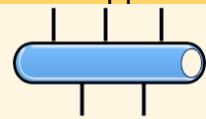
**Articulation:**  
SW Creation Needs



**Process:** Container

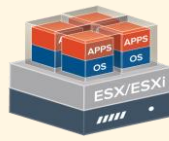


**File:** Blueprint



**Signal:** Overlay

**Operation:**  
VIM= Virtual Infrastructure Manager



**Run:** Software Defined Computing



**I/O:** Software Defined Storage



**Connect:** Software Defined Network



**Computer Type**



**SW Creation: Object Oriented Architecture**  
Computing Languages Run-Times



**SW Ecosystem**  
Desktop of Monolithic Applications

**Articulation:**  
SW Creation Needs



**Process:** Process



**File:** File



**Signal:** Buffer

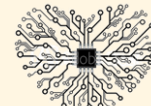
**Operation:**  
Hardware Manager



**Run:** Memory



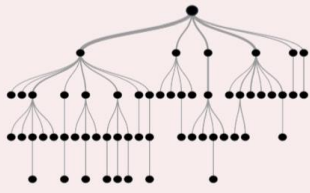











**I/O:** Devices, Disks



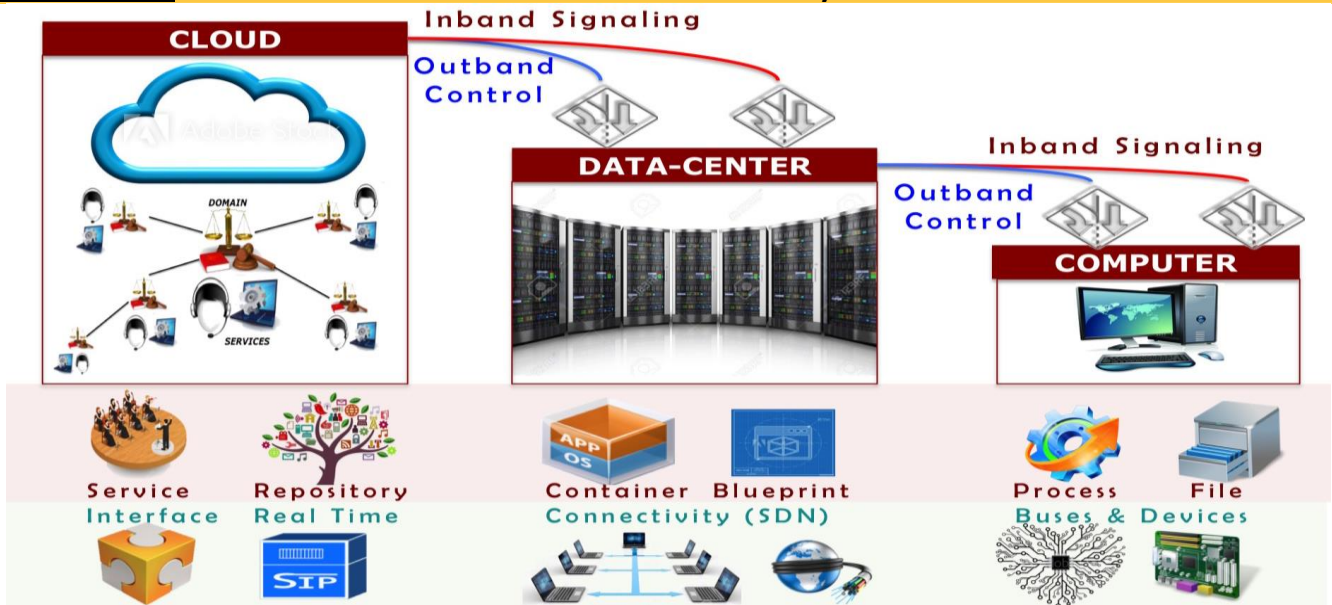
**Connect:** Buses

**ARTICULATION: SW CREATION BASIC NEEDS - COMPUTING OPERATING SYSTEM**

	<b>CREATION</b>	<b>MANAGEMENT</b>	<b>EXECUTION</b>
 <b>Algorithms</b>	<b>Kernel Orchestration</b> 	<b>Tree Organization</b> 	<b>Authentication and Authorization</b> 
 <b>Values</b>	<b>Abstraction;</b>  <b>Process</b>  <b>File</b>	<b>Outband Control</b>  <b>PID, PPID Execution</b>  <b>Update Packaging</b>	<b>Inline Signalling</b>  <b>Credential</b>  <b>Command</b>  <b>Signal</b>

Computing is about creating programs, code is placed in files while executed in processes, therefore software creation requires to massively articulate file and process abstractions through operating system scheme of roles:

- **Creation – Spawn Values:** kernel, or system core, spawn process and file values. The architecture can be located in operating systems bibliography, such “Solaris Internals” from Jim Mauro and Richard McDougall.
- **Management – Choreograph all Values - Outband Control:**
  - *Central Control...* processes and files are organized in tree structures through ID and parent ID keys.
  - *Update...* values provisioning (ie.: SW repository systems).
- **Execution – Orchestrate each Value Life Cycle - Inline Signaling:** values require authentication and authorization during the different stages of the execution:
  - *Credential...* the identity problem.
  - *Command...* shell to launch applications.
  - *Signal...* run-times, cooperation schemes between processes to stand programming languages, such Java or any other.

**OPERATION: TYPES -THREE AGGREGATION LEVELS, STRATIFICATION PROCESS.**

According to principles of harmony, each type should implement the very same scheme of roles to get different types of compact units for flexibility purposes.

➤ **At Computer System level, the final handling of the application.**

- Operation Infrastructure: devices, peripherals and buses of each unit. The operating system offers tools to control buses and devices.
- Values: process and file values. Memory is used as middleware for signal exchange between processes.
- Execution: users identity at operating system and application execution levels.
- SW Creation: OOA- Object Oriented Architectures, Computing Language Run-Times, such Java Run-Time.

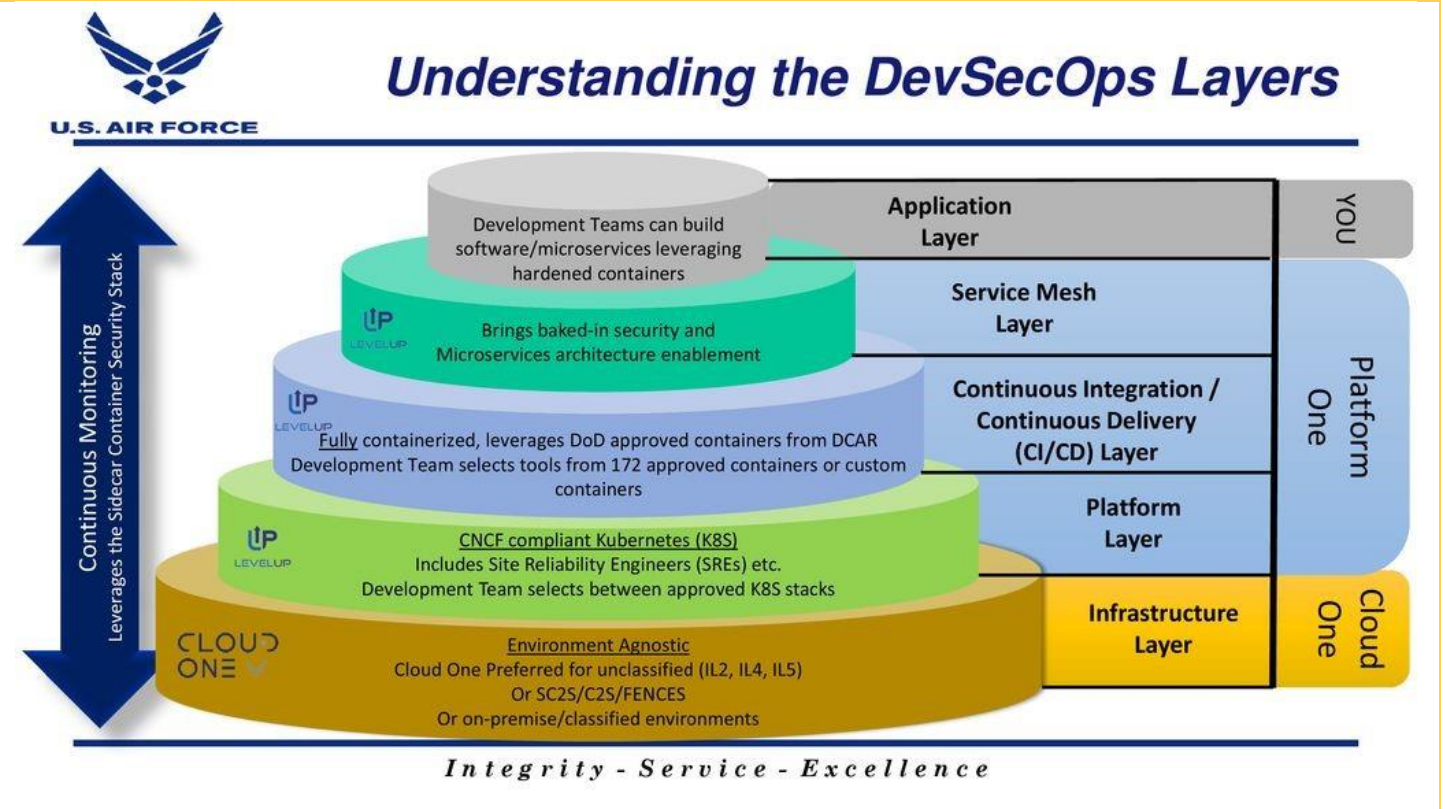
➤ **At Data-Center System level, deployment in each area**, install services and repositories in a secured and hermetic manner.

- Operation Infrastructure: connectivity between computers and the access connectivity to the group of computers is the operation infrastructure that enables the group of machines to work as a compact unit. Virtualization and Software Defined Network tools control the operational infrastructure
- Values: container and application blueprint. A virtualized network (or Overlay) work as middleware for signalling exchange between containers.
- Execution: users are centrally synchronized across the data-center and aggregated in tenant's identities for access permissions to containers and zones levels (groups of applications with associated assets and networking policies): data-center micro-segmentation strategies.
- SW Creation: SOA- Services Oriented Architectures, Micro-service Run-Times that containerize OOA inside API and data structure and introduce it into cooperation machinery between containers.

➤ **At Cloud System level, distribution through areas** controlling threads of services and repositories to apply, growth and accessibility schemes to the data, as well as cloning and cooperation schemes between logics.

- Operation Infrastructure: distributed systems operation infrastructure enables to stand distributed processes and repositories.
- Values: service and repository. An RTP network (similar to voIP) is used as middleware for signal exchange between services.
- SW Creation: DOA - Distributed Oriented Architectures, Scalability Run-Times that pack SOA into an interface to create threads across data-centers (ie.: Nir Simionovich, scaling voIP: <https://youtu.be/Sk3nuKMB7o8> ).



**DATA-CENTER TYPE: STATE OF ART, SCATTER PROJECTS.**

**Articulation:** Cluster application basic needs.

- ✓ Life cycle Orchestration:
  - *Creation:* Companies such D2IQ are working in Data-Center Operating System, they already find out kernel layering structure with associated roles and responsibilities, but implemented through incomplete open source projects.
  - *Execution:*
    - *Credential:* projects such *Aporeto* are working on identity problem.
    - *Command:* projects such *Ansible* are addressing this challenge.
    - *Signal:* bus of events that run on top of overlay networks, projects such Kafka, RabbitMQ, etc.
- ✓ Applications Choreography: coherent applications working environment across data-centers. Projects such Pivotal, OpenShift, Deis, Cloud Foundry or Heruku are working on that.

**Operation:** Data-center resources handling.

- ✓ Roles deployment: software defined compute, storage and network. Companies such Arista and NetApp are working on that.
- ✓ Roles control: container life cycle orchestration to massively be used in the creation of applications. Projects such Kubernetes or Mesos for container handling, while OpenStack for data-center control.



### Social System

Connection between Services

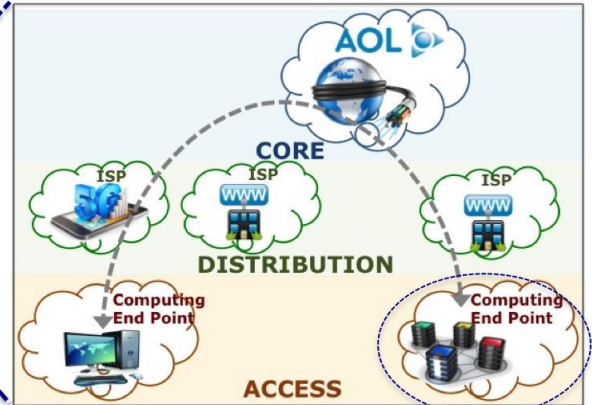
Examples of ecosystem services in the Netherlands



### Internet System

Ecosystem: Autonomous Systems

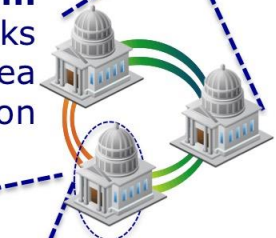
Creation: End to End Connection



### Autonomous System

Ecosystem: Networks

Creation: Wide Area Connection



### Network System

Ecosystem: Devices

Creation: Service Area Connection



# Network Infrastructures.

## Information Distribution



**Internet System Type**



**Connection Creation:**  
Across a Network of Autonomous System



**Connection Ecosystem**  
Internet Registry Systems

**Articulation:**  
Connection Creation Needs



**Core:** Multihomed Autonomous System



**Distribution:** ISP Autonomous System



**Access:** Computing Autonomous System

**Operation:**  
Legal Infrastructure



**Node:** Autonomous System



**Link:** SLA Contract, BGP

**Autonomous System Type (AS)**



**Connection Creation:**  
Across a Network of Service Areas



**Connection Ecosystem**  
AS Operation Center, Software Define Infrastructure

**Articulation:**  
Connection Creation Needs



**Core:** Wide Area Network (ie.: MPLS)



**Distribution:** Transport Network (ie.: SDH)



**Access:** Service Area Network (ie.: IP)

**Operation:**  
Inter-Network Infrastructure



**Node:** Network



**Link:** Inter-Network Protocols

**Network System Type**



**Connection Creation:**  
Within a Service Area



**Connection Ecosystem**  
NOC=Network Operation Center, Software Define Network

**Articulation:**  
Connection Creation Needs



**Core:** Traffic Engineering (ie.: routing)



**Distribution:** Traffic Aggregation (ie.: switching)

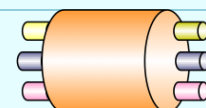


**Access:** Media Share (ie.: RAN, UTRAN, LAN)

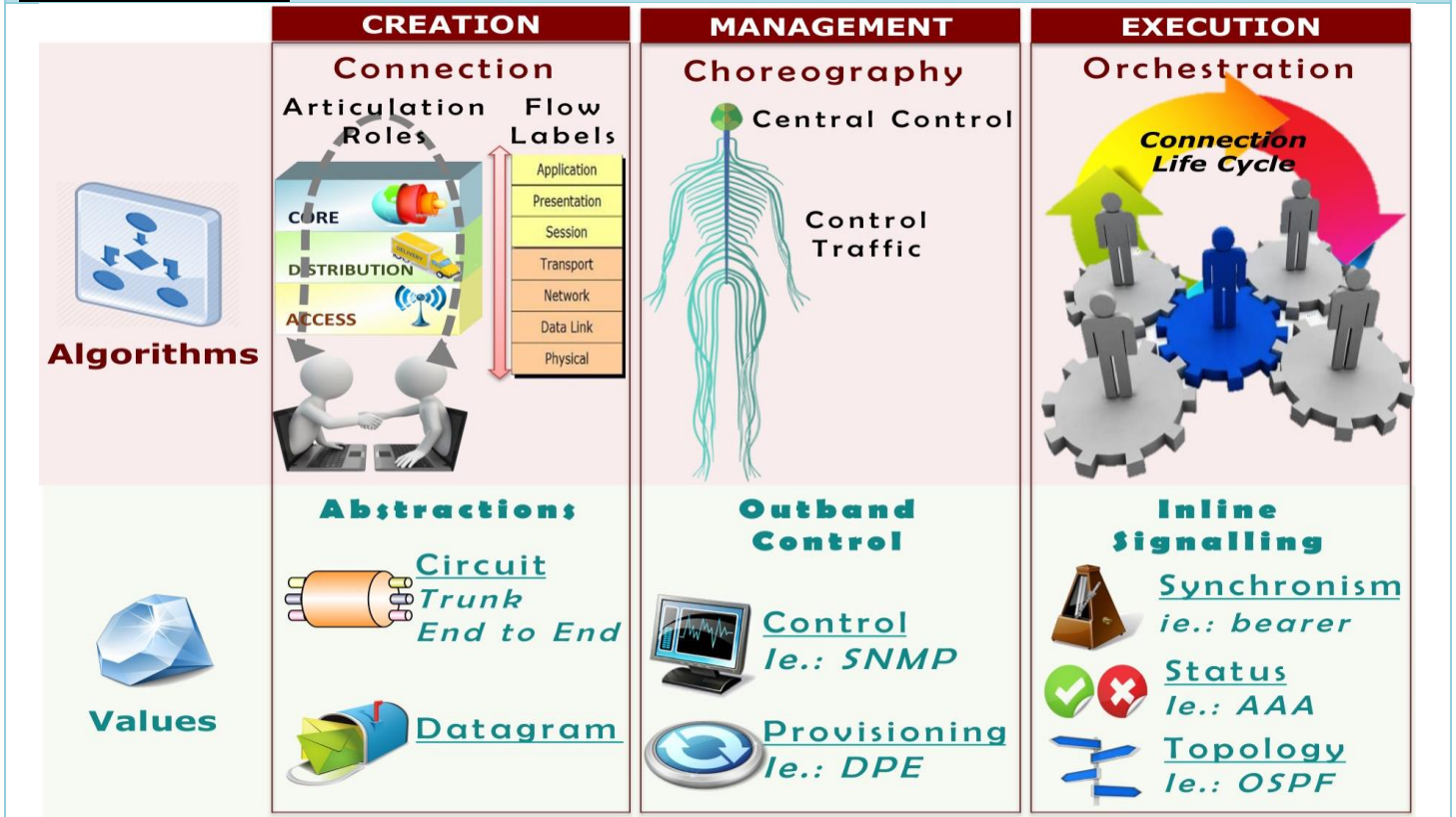
**Operation:**  
Network Infrastructure



**Node:** Device

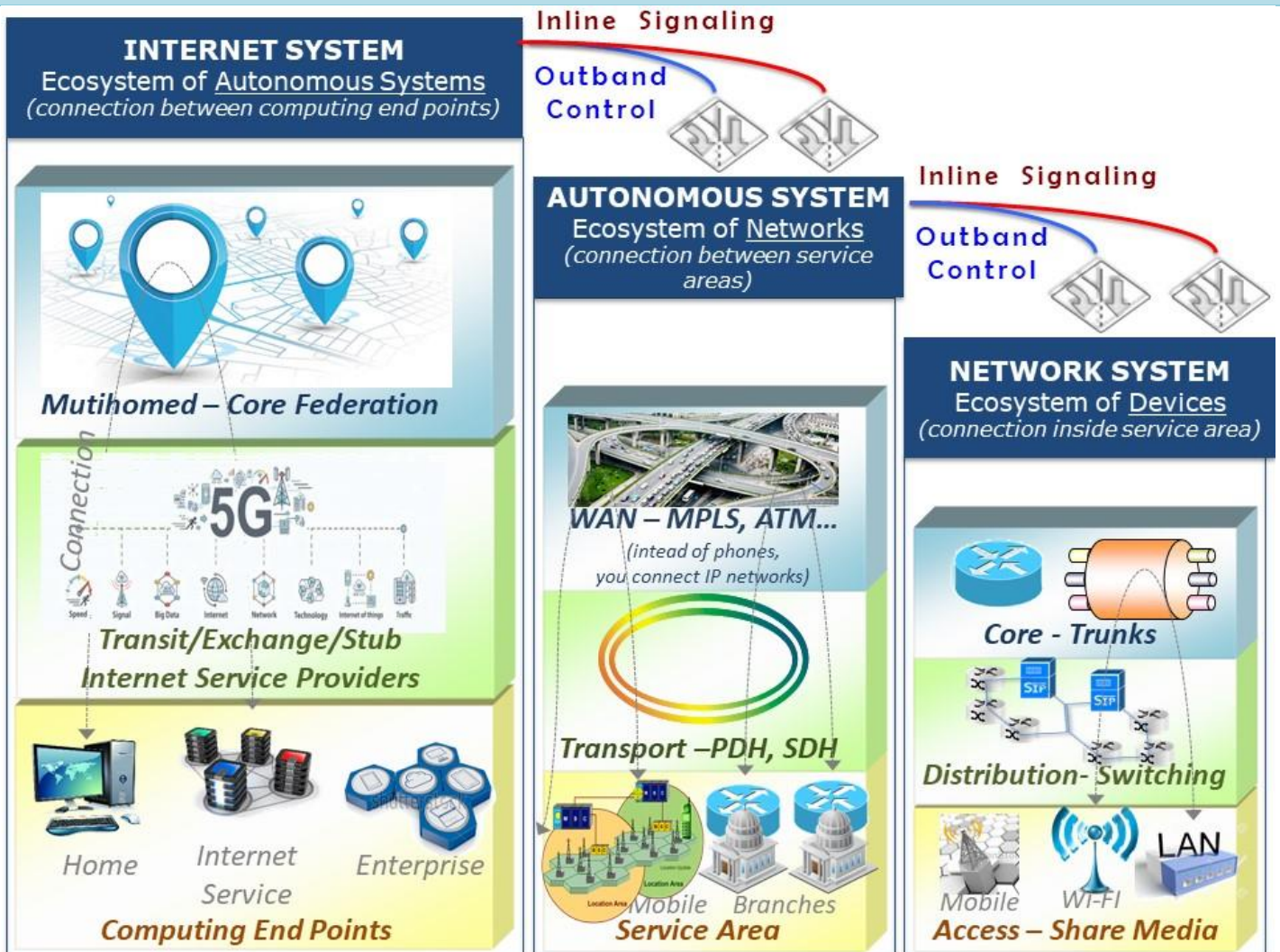


**Link:** Inter-Device Protocols

**ARTICULATION: CONNECTION CREATION BASIC NEEDS - NETWORK SYSTEM**

Networking is about creating connections massively articulated through core-distribution-access scheme of roles:

- **Creation – Spawn Connections:** core-distribution-access scheme massively creates datagram for volume (similar to postal service) and circuit for real time (similar to phone service) with associated labeling protocols (OSI stack as reference model, but each type develop his own stack):
  - *Core:* federation of service areas with distributed systems algorithms.
  - *Distribution:* traffic aggregation in a hierarchical scheme up to three levels. In phone networks, PDH for copper, SDH for optical networks.
  - *Access:* sharing media access across users with different technics.
- **Management – Choreograph all Connections - Outband Control:** management models can be found in bibliography such "*Integrated Management of Networked Systems*" from Heinz-Herd Hegering, Sebastian Abeck and Benhard Neumair. In phone networks it is called OSS (Operation Support Subsystem).
  - *Control Traffic...* (ie.: SNMP, openFlow) that allow central control (ie.: EMS/NMS, controller based IP networking).
  - *Provisioning...* updates across devices (ie.: user registration).
- **Execution – Orchestrate each Connection Life Cycle - Inline Signaling:** connection requires cooperation between devices thanks to signaling protocols (ie.: SS7 between switches and session headers, IP routing protocols). In phone networks is called BSS (Business Support Subsystem):
  - *Devices Synchronism...* media share (ie.: bearer channel, collisions, errors, MISI, watchdog, keep alive).
  - *Connection Status...* connection life cycle control (ie.: AAA, billing, QoS).
  - *Network Topology...* visibility to coordinate operations (ie.: OSPF, ARP).

**OPERATION: TYPES -THREE AGGREGATION LEVELS, STRATIFICATION PROCESS.**

Russian doll fashion aggregation effect (red and blue lines between types in the picture) means: a) core service area circuits (ie.: Ethernet VPLS) are packed in wide area network circuits (ie.: MPLS); also b) software defined data-centers with associated repeaters and networking hardware are packed for monitoring from a single autonomous system network operation center:

- **At Network System level, service area deployment**, an ecosystem of devices.
  - Operation Infrastructure: devices and links, software define network (controller based networking) to control operation infrastructure.
  - Values: circuit and datagram values.
  - Connection Creation: SAC=Service Area Connection between devices.
- **At Autonomous System level, connection across service areas**, an ecosystem of networks to create a social entity, such company or service provider.
  - Operation Infrastructure: service areas (such IP networks) and trunks between them, network operation centers to control operation infrastructures.
  - Values: trunk circuits and datagram containers values.
  - Connection Creation: WAC=Wide Area Connection between service areas.
- **At Internet System level, connection between computing end points**, an ecosystem of autonomous systems.
  - Operation Infrastructure: autonomous systems and connections between them.
  - Values: circuit and data compliance agreements between autonomous systems.
  - Connection Creation: E2EC=End-to-End Connection between computing systems.