Integrated Communication Systems

http://www.icsy.de

Future Internet a Service oriented Approach

SONATE

(Service-oriented Network Architecture)

7th Würzburg Workshop on IP: Joint EuroFGI and ITG Workshop on "Visions of Future Generation Networks" (EuroView2007)

Paul Müller, Bernd Reuther, AG ICSY, University of Kaiserslautern, http://www.icsy.de

Content

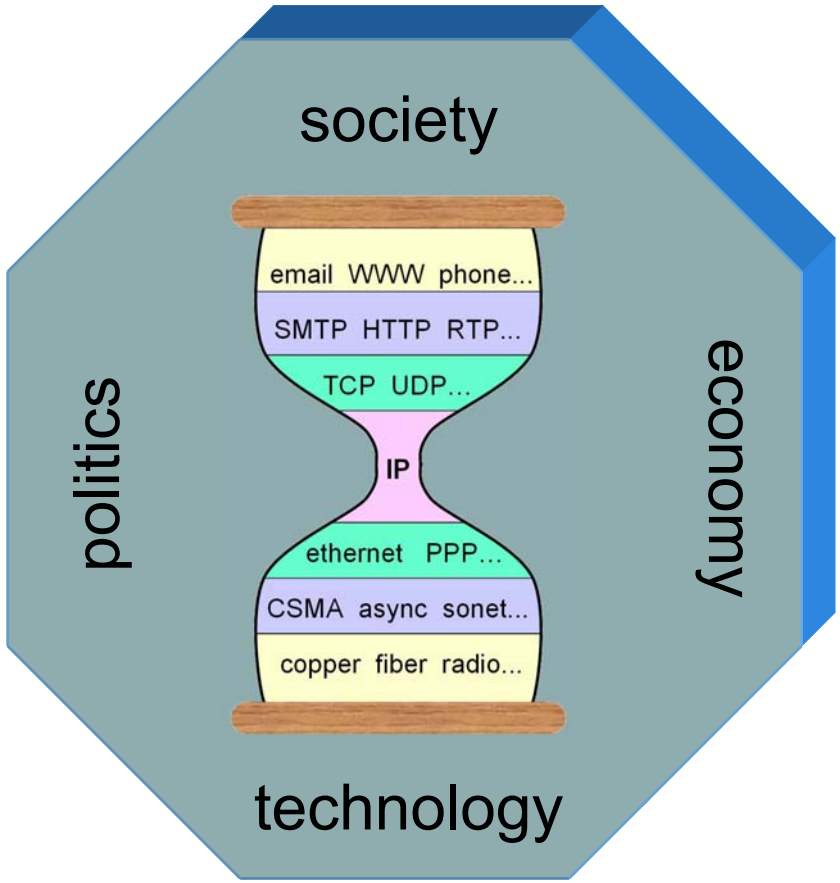
- What we are talking about
- Related Work
- Service Approach for the Future Internet
- Conclusion

1.

What we are talking about

http://www.icsy.de

What we are talking about?



- I do not talk about:
  - Society
  - Technology
  - Economy
  - Politics

  } **STEP**
- But: Architecture and how it can evolve a new “inter-networking” Paradigm
  - At least as good as todays solution
  - Replace the traditional protocol layering paradigm with a more general model

Architecture

- Generic definition of the term architecture:
  - The art and science of designing structures
- In computer science, architecture is the (ANSI/IEEE Std. 1471-2000):
  - fundamental organization of a system
  - relationship of components (to each other and environment)
  - design and evolution principles
- Why is architecture a challenge?
- Question for dynamic software systems?
  - Which and how much system specific information (functionality, environment, usage, ...) should be considered by an architecture ?
    - too little information cause unstructured or even chaotic systems
    - too much information cause inflexible systems

http://www.icsy.de

Architecture of the current Internet

- Fundamental organization
  - Layered Structure
  - One or more parallel protocols per layer
  - Functionality per layer is defined and fixed by a model (OSI or TCP/IP).
  - Location of functionality (end-system or network) motivated by the “end-to-end argument”
- Relationship of components
  - Each layer uses services of lower layers and offers another service to the upper layer
  - Interfaces between layers are not defined, only few common interfaces exist, most prominent:
    - The (Berkley) Socket Interface (access to layer 4)
    - NDIS Network Device Interface Specification (access to layer 2)
  - Interface between protocols of the same layer are not defined (IP ↔ ARP, IP ↔ Routing-Protocols)
- Design and evolution principles
  - Overall
    - It should be possible to redesign a layer and its protocols without having to change the adjacent layers (OSI specification)
      - But IPv6 requires a new TCP implementation
  - Per layer
    - Use only services of lower layers, i.e. mechanisms of lower layers are transparent
      - But TCP/UDP include IP-pseudo-header in CRC
  - Per protocol
    - Options
    - Version numbers
    - Some bits for “future use”

http://www.icsy.de

Basic assumptions … but

- The end-to-end principle
  - Do not trust the network
    → implement several functionalities in end-systems
  - There are many miss behaving / malicious end-systems (users)
    → today the network can not trust the end-systems
- Layered Architecture
  - Transparency of lower layers → simplifies design
  - Today we have cross layer design – security balconies- …
- Keep the network simple
  - Stateless IP in the core → robustness
  - today we have additional mechanisms for QoS / MPLS …
- Convergence is based on IP
  - Widespread and mature technology
    → available within many devices (from PDA to HPC)
  - Widespread technology with low potential for evolution
    → mobility, sensor networks, …

http://www.icsy.de

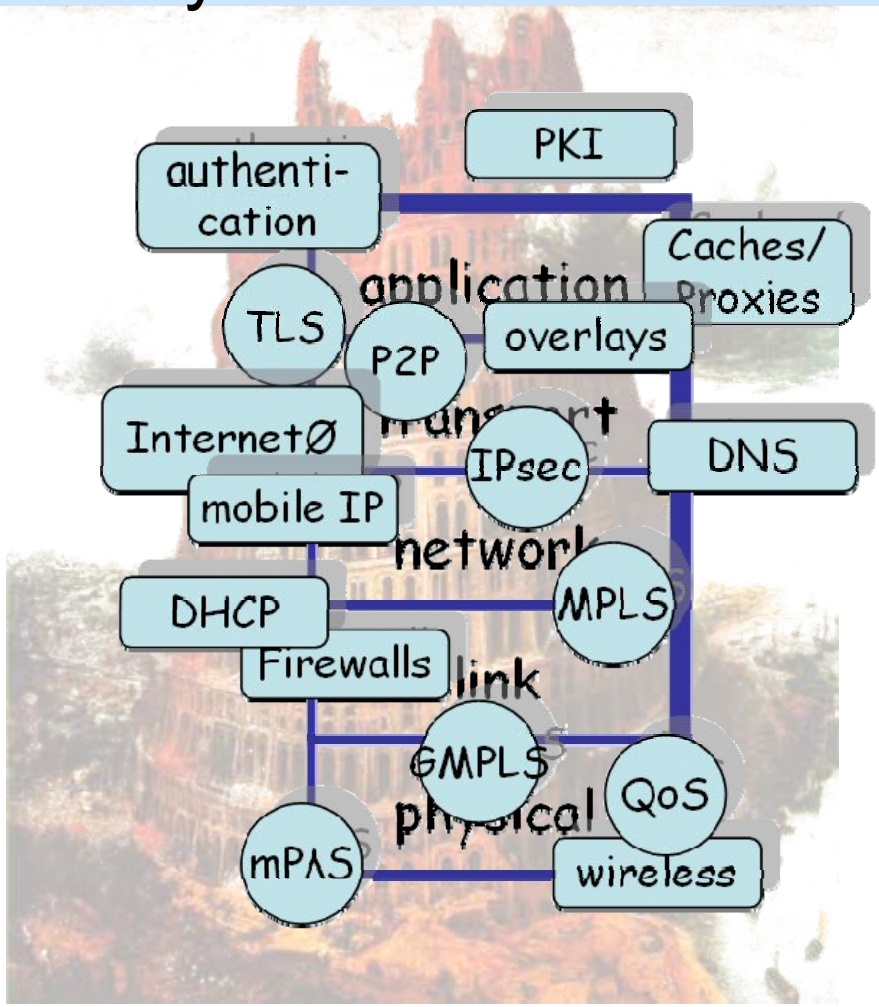
Problems of the current Internet

- Low degree of flexibility
  - Short term:
    adaptivity and adaptability according to environmental conditions and user requirements
    - no negotiation of capabilities
    - Inflexible protocols
  - Long term:
    enhance and exchange functionality
    - Exchange nearly impossible (e.g. IPv4 -> IPv6)
    - Enhancements in narrow bounds is possible
- Typical solution today:
  - Cross-layer optimization:
    improve adaptivity / adaptability
    - Optimize several protocols
    - Violate layered structure
  - Overlay networks:
    new mechanisms
    - Rebuild functionality of "lower layers“ at “higher layers”
    - Enables (new) functionality for few applications only
      - How many overlays will be required?

http://www.icsy.de

Architecture of the current Internet summary

- Original architecture is violated
  - Middleboxes (NAT, caches/proxies,…)
  - Intermediate layers (TLS, IPsec, MPLS, ...)
  - Specialized network domains (areas with specific QoS or security properties)
- Increasing interdependencies hinder innovation
  - Hard to integrate new mechanisms
  - QoS / CoS
  - Mobility
  - Security / Authentication
- Complexity is still rising …



http://www.icsy.de

but …

There are fundamental issues with the current architecture and many of its mechanisms that cannot be fixed incrementally with additional engineering workarounds.

Peter A. Freemann

http://www.icsy.de

2.

Related Work

http://www.icsy.de

Related Work 1

- The most interesting clean slate approaches which are related to the work presented here can be seen in the
  - role-based approach (RBA) conducted in the NewArch project;
  - and the SILO approach introduced by Dutta et. al.;
  - And more general the Clean-Slate project at Stanford.
- The RBA represents a non-layered architecture organizing communication in functional units referred to as “roles”.
  - Roles are not hierarchically organized, and thus may interact in many different ways;
  - as a result, the metadata in the packet header corresponding to different roles form a “heap” not a “stack”, and may be accessed and modified in any order.
  - The main motivation for RBA was to address the frequent layer violations that occur in the current Internet architecture

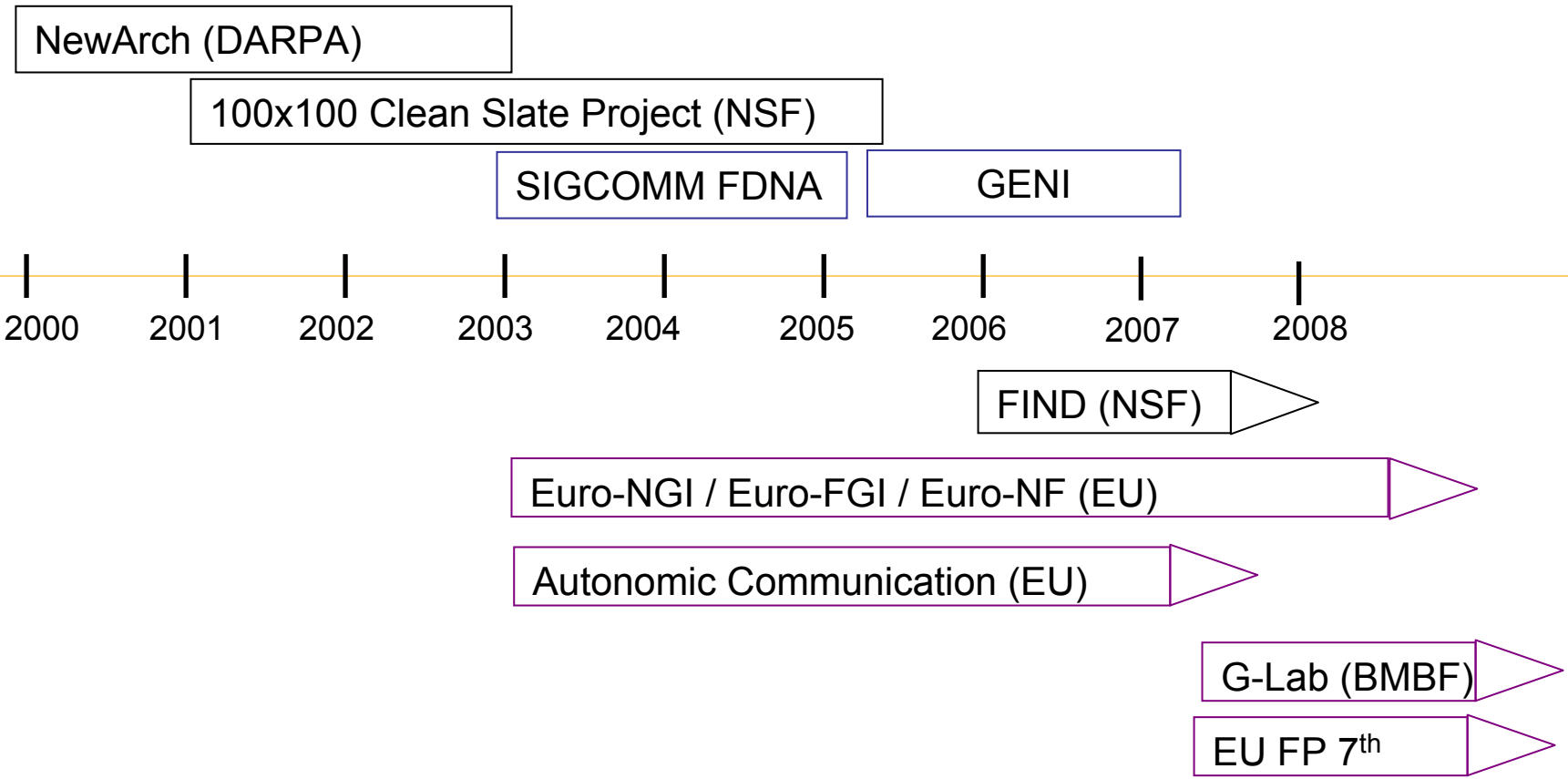
http://www.icsy.de

Related Work 2

- The SILO approach also introduces a non-layered design
  - It is based on silos of services assembled on demand and specific to an application and network environment.
  - it offers a more flexible header structure than the RBA approach
  - The overall goal of the SILO architecture is to facilitate “cross-layer” interactions
- Some earlier work in the area of micro protocol architectures investigated also more flexible communication frameworks.
- SONATE, RBA and SILO are similar approaches; all three avoid layering and aim to define a highly flexible architecture.
  - The main motivation for RBA and SILO was to address the frequent layer violations that occur in the current Internet architecture. Our approach is a more holistic one triggered by service-oriented architectures in the application domain.

http://www.icsy.de

Related Work 3



3.

Service Approach for the Future Internet

http://www.icsy.de

Service Approach for the Future Internet

- Basic Idea:
  - A communication system made of loosely coupled services
    → avoid implicit premises as much as possible
  - Apply SOA principles to communication systems (requires new techniques)

- Define explicit interfaces and interaction between elements of the architecture
  - Dependencies to each other

- Explicitly refer to required/offered functionality and data structures
  - Enables change of functionality and data structures and thus provides higher degree of flexibility

The term “Service” 1

- A definition from business economics:

Service is a customer-oriented result. This result is produced when an organization performs activities that are oriented towards meeting customer needs and expectations. [ISO 9000]

  - A service is a result (or benefit) for a customer
  - A service involves activities
  - A service involves two parties: provider and customer (which have to communicate somehow)
- How to describe a service ?
  - By action: can be very precise but requires specific knowledge
  - By what’s communicated: interface must be defined nonetheless but these often lack semantic
  - By benefit: is independent of activity and thus fosters loose coupling, but requires an expert to find an appropriate provider

The term “Service” 2

- A unit of work done by a service provider to achieve desired end results for a service consumer. The results of a service are usually the change of state for the consumer but can also be a change of state for the provider or for both

- Benefit: Loose coupling between the participating software agents, enabled by:
  - A small set of **simple and ubiquitous interfaces.**
  - **Descriptive** messages constrained by an extensible schema delivered through the interfaces. None, or only minimal, system behavior is prescribed by messages.
  - **Extensibility**
  - **Service discovery**

- Focus on exposing business functions, not technology

http://www.icsy.de

Lessons learned from SoA 1

- 1. *Use self-contained services and data*.
  - This means to find appropriate granularity of services and data.
- 2. *Use explicit descriptions instead of implicit assumptions* (fostering loosely coupled services).
  - The current Internet is based on several "well known" protocols.
    - But presupposing protocols (i.e. rules and formats) implicitly lay down several technical details which are hard to change afterwards.
  - As implicit assumptions raise the grade of coupling, we propose to use explicit descriptions where ever possible:
    - a) explicitly describe required services, e.g. "forwarding of data", "reliable transmission" or "flow control";
    - b) explicitly describe types of data.

Lessons learned from SoA 2

- 3. *Use well defined interfaces between services* (fostering loosely coupled services).
  - Each service must have a well defined interface, hiding its internal mechanisms and data structures.
  - Protocols are the building blocks of the current Internet.
- But interfaces between different protocols – if they exist at all – do not hide protocol mechanisms. For example the BSD Socket Interface does not hide which transport or network protocols are used, this hampers exchanging protocols because adjacent layers also require adaptation.

http://www.icsy.de

Services Categories 1

- Infrastructural services:
  - Services on the lowest level represent the provisioning of resources
    - Like Ethernet, Frame-Relay or ATM offer infrastructural services, but also P2P overlay networks, ad-hoc networks, or even sensor networks offer such services.
- Basic communication services:
  - This category offers the transport of data between two or more entities.
    - Therefore functionalities like switching, routing, and signalling must be implemented. Each of them, may be individual services. Today protocols like TCP/IP, ATM, ISDN and even H.323 and SIP implement basic communication services among other things.

Services Categories 2

- Extended communication services:
  - Examples for these are privacy, security, availability, robustness, accounting, and quality-of-service,
  - The necessity of these services depends on the demands of consumers and providers.
- Intermediary services:
  - Services that help in finding appropriate services.
  - Bootstraping systems
- Context adaptive and intelligent user services:
  - These are types of services providing users the ability to access the services they need, anywhere and from any device.

http://www.icsy.de

Services Categories 3

- Information services:
  - Services that provide (personalised) information, for instance by comparing, classifying, or otherwise adding value to separate information sources.
- Mangement services:
  - All kinds of services have to be managed. This covers tasks like service deployment, monitoring, (self-) configuration, and (self-)optimization.

http://www.icsy.de

Examples of Services 1

- Error handling
  - Error detection
    - CRC, Hash, ...
  - Error notification
  - Error recovery
    Retransmission or FEC
- Identifier
  - Identify an endpoint
  - Identify a location
  - Identify a process
  - Identify a user
- Multiplexing
- Fragmentation/Assembly
  - Fragmentation
  - Segmentation
  - Blocking
- Connection / flow setup
  - Implicit
  - 3-way handshake (e.g. TCP)
  - 4-way handshake (e.g. SCTP)
  - Dynamic label distribution (e.g. MPLS/GMPLS)
  - Halfclose
- Address resolution
  - ARP, DNS
- Routing / forwarding
  - Use local routing tables, DHT
- Flow Control
  - with respect to destination
  - Congestion control, i.e. with respect to network
  - Rate control
- QoS / CoS
  - Classes & Aggregation
    - DiffServ
  - Signaling
    - RSVP
- Path management
  - Path-switching
  - Path monitoring
    - keep-alive, heartbeat
  - MTU Discovery
- Multicast
- AAA
  - Authentication
    - 802.1x, Radius, TACACS, Kerberos
  - Authorization
  - Accounting
- Encryption
  - Key Exchange
    - Diffie-Hellman, RSA
  - Cipher-Algorithm
    - DES, 3DES, AES
- Real Time support
  - Content identification
  - Source identification
  - Start / Stop marker
  - Time + Sequence number
- Communication patterns
  - Request / Reply
  - Message Passing
  - Message Queuing
  - Publish / Subscribe
  - Media Streaming
  - File transfer

1. *This list is not intended to be complete*
2. *The protocols mentioned are not services by themselves, they are only examples for mechanisms*

Used by / required for TCP/IP

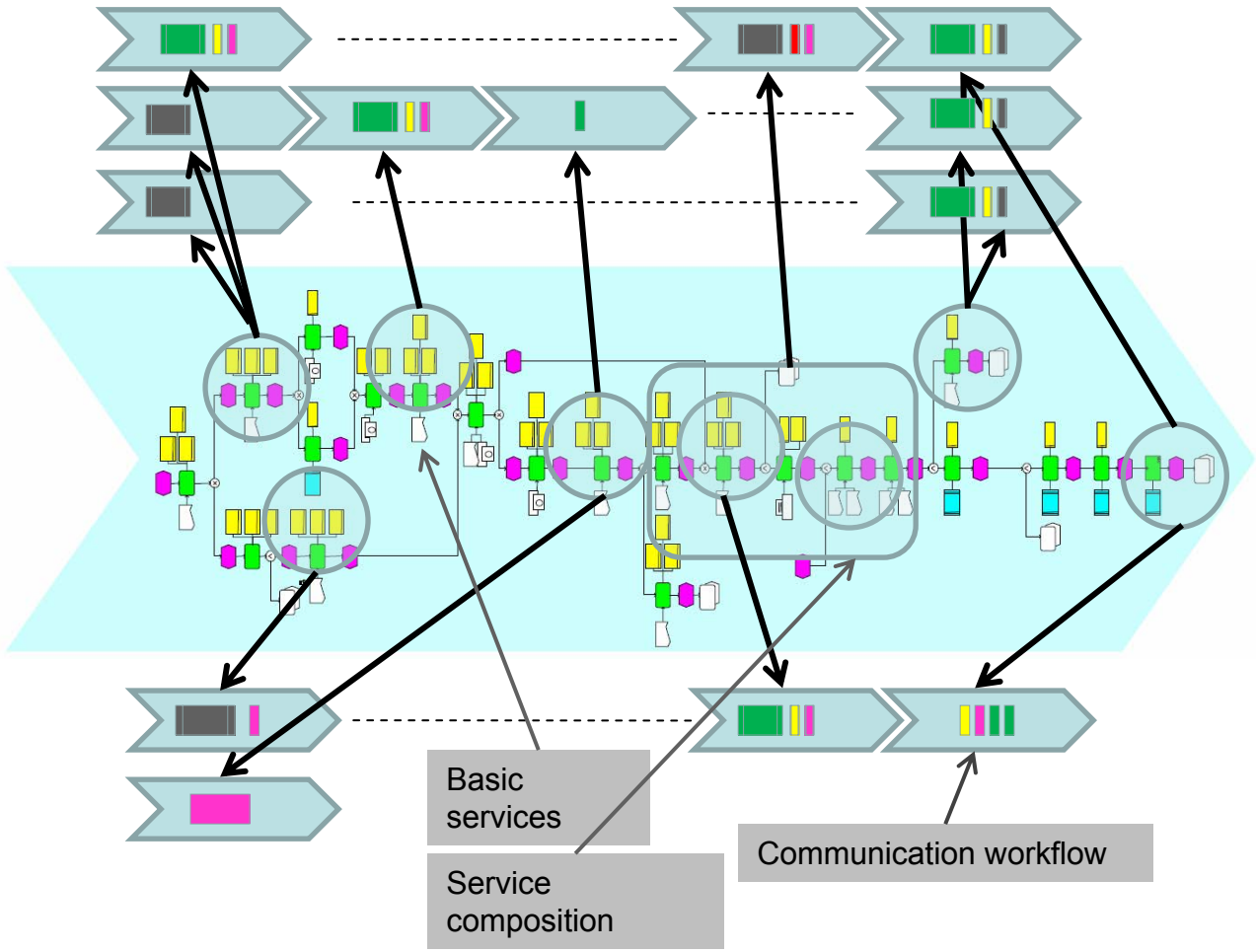
Examples of Services 2

- Loop detection and elimination
  - STP, MSTP, RSTP
- Trunking
- Virtual path / tunneling
  - MPLS/GMPLS
- VLAN tagging
- Load balancing
- Routing / determine topology
  - IS-IS, OSPF, IGRP, RIP
  - BGP
- Monitor infrastructure
  - Load
  - Error rates
  - Signal strength (wireless)
  - Availability
- Traffic engineering
- Network Management
  - Get / Set (e.g. SNMP)
  - XML based (e.g. netconf)
  - Provide configuration data
    - DHCP, TFTP
- Capability negotiation
- Network admission control
  - by user
  - by device / device configuration
- Network protection
  - Firewalls
  - Intrusion Detection
- Resilience
  - Path/Node failure
- Self-organization and self-management techniques

1. *This list is not intended to be complete*
2. *The protocols mentioned are not services by themselves, they are only examples for mechanisms*

http://www.icsy.de

Service Approach for the Future Internet

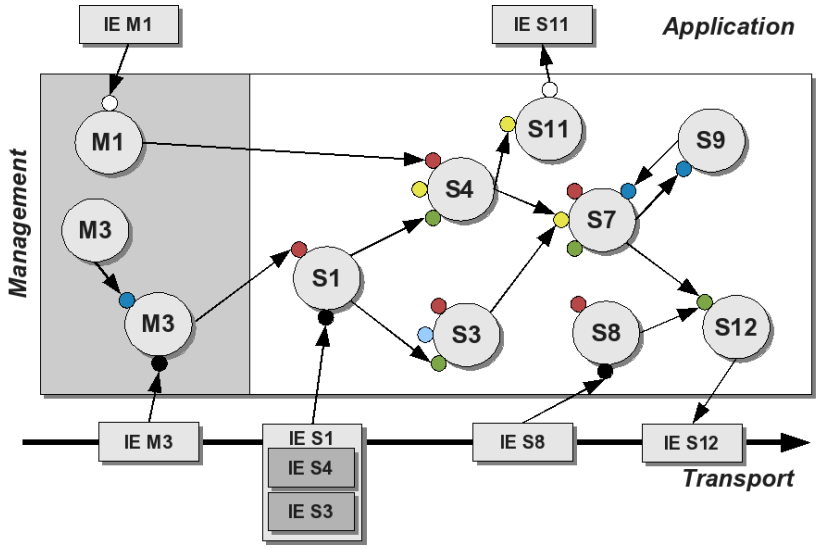


Decomposing protocols into a set of reusable service elements that can be recomposed in different ways depending on application and network properties

http://www.icsy.de

Illustration of the Services Concept

- Metadata increase flexibility
  - Explicit references to services
    - Simplifies provision of new services
  - Explicit descriptions of data types
    - Simplifies extensions of mechanisms (add optional data)
    - Enables alternative mechanisms (add alternative data)
- Services have well defined interfaces
  - Enable exchange of service implementations
- Services are fine granular
  - Similar to micro protocols
- Similar to “role based architecture” approach but:
  - Roles can be exchanged/replaced, but it is not possible to extend roles (e.g. add optional data)
  - Separates application payload from protocol header, i.e. one role can not contain other sub-roles.



http://www.icsy.de

The Service Approach 1

- Avoid complex protocols
  - There is no need to bundle functionality that might be used independent of each other
  - Protocol decomposition to micro protocol is not new, e.g.
    - Dynamic Network Architecture (O'Malley & Perterson)
    - Dynamic Configuration of Light-Weight Protocols (Plagemann, Plattner, Vogt, Walter)
    - Componentized Transport Protocols (Condie et al.)
    - …
- The service approach is more general
  - Replacing ***implicit assumptions*** by ***explicit references*** does not reduce functionality

http://www.icsy.de

The Service Approach 2

- Avoid to presuppose where some functionality is placed (end-system, network or network domain)
  - The end-to-end argument postulates that some functionality can only be implemented in end systems. But is the location of a functionality a principle that never changes?
    - *Saltzer, Reed, and Clark mention an alternative to end-to-end implementation: The goal would be to reduce the probability of each of the individual threats to an acceptably small value. This was considered to be too uneconomical (1984) → is this true today and in future ?*
    - *Moors argues that the end-to-end argument is mainly derived from trust and not from technical issues → what is acceptable depends on requirements!*
    - *Typically reliability should be provided end-to-end, but interceptions of TCP connections by proxies are reality today. Who cares about the reduced reliability?*
  - The architecture should not presuppose where some functionality is located, because this may change (but an application may do so).
  - In consequence: a layered structure is no longer appropriate

http://www.icsy.de

Architectural Issues

- Flexibility
  - in the long run no fixed set of mechanisms/protocols will fulfill all requirements and are appropriate in all environments
  - Consequence: the future Internet must be flexible according to the mechanisms used
- Scalability
  - The future Internet should be accessible for everybody, everywhere, every time and at every scale: Dimension , Capabilities of links, Capabilities/resources of nodes
- Application neutrality
  - Do not presuppose who will use the network and how
    - Note: the current Internet was originally developed for data exchange between computers only

http://www.icsy.de

4.

Conclusion and Summary

http://www.icsy.de

What we have learned from Service Orientation

- Eight common principles
  - Reusability
  - Formal contract
  - Loose Coupling
  - Abstraction
  - Composability
  - Autonomy
  - Statelessness
  - Discoverability
- Demand and foster a new thinking (“think SOA”)
- SOA is not a fixed solution, it’s a process
- Services should be
  - Solution-agnostic
  - Entity-centric
  - Not task-centric
- Service composition enables
  - Flexibility
  - Evolution
  - Dependability
  - Individual workflows

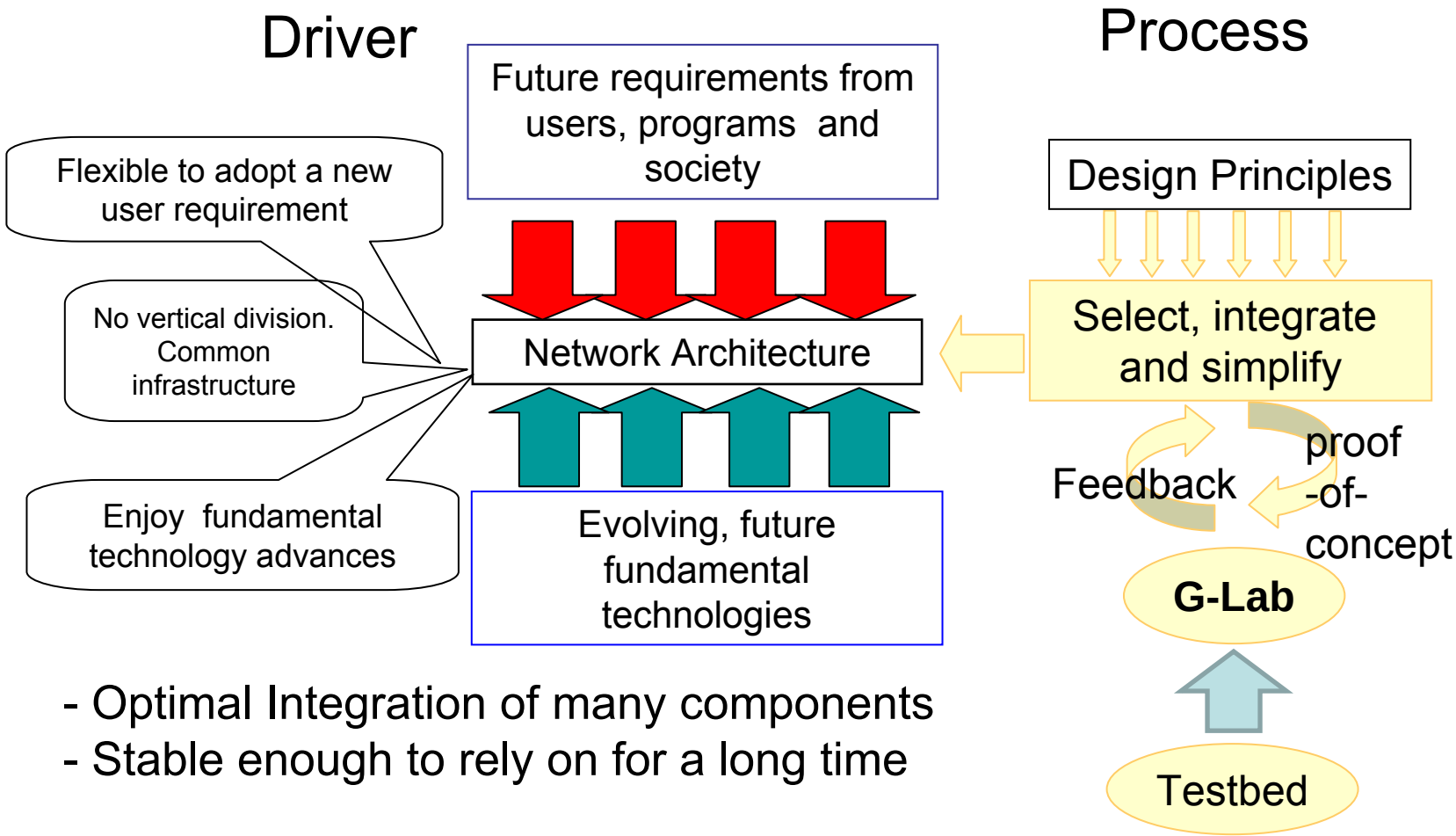
http://www.icsy.de

Advantage

- For users
  - Adaptivity / Adaptability to environment
    - optimized performance
  - Adaptivity / Adaptability to requirements
    - optimized qualitative properties (i.e. QoS)
  - Request services instead of mechanisms
    - Easy to use, because much less technical know-how required
  - Extendable set of mechanisms
    - Large toolbox of services available
- For providers
  - Extendable set of mechanisms
    - Add functionality needed locally (e.g. for traffic engineering, accounting, management, …)
    - Easy to deploy new services
  - Reduced dependencies between mechanisms
    - Improved robustness

http://www.icsy.de

Conclusion



- Optimal Integration of many components
- Stable enough to rely on for a long time

Summary

- Tight coupling in the current Internet hinders adaptivity as well as evolution
- Goal: an architecture of loosely coupled elements (services) suitable for a future internet

Convergence

Should not be based on protocols

but on

Architecture

see also: http://www.future-internet.org

http://www.icsy.de