
FIA Panel: Future Internet Architectures
Poznan 26/10/2011
We can't solve problems by using the same kind of thinking we used when we created them.
The Internet evolution...

... but over time...

Paul Mueller, University of Kaiserslautern
Problem statement

Problem:

It is hard to integrate new functionalities into the current Internet.

Cause:

- Lots of implicit dependencies, i.e., tight coupling.
- The problem is not limited to specific protocols or mechanisms.

We assume the Internet as a "largely distributed software system".

It is an architectural issue!
Basic idea: apply SE methods for designing a new software architecture for the Internet core:
- Apply SOA principles* to communication systems (requires new techniques)
- A communication system made of loosely coupled services (functionalities)

A communication system based on the SOA paradigm:
- Service should be self-contained
- Define explicit interfaces and interaction between elements of the architecture
- Minimize assumptions about other services

Dynamically interacting services replace the concept of layers

* Don't equate SOA with Web Services. SOA is an approach to designing systems, Web services is an implementation technology.
Software defined Networks: Flexibility is the Key

- Long term flexibility:
  - the capability of a system to evolve with updated protocols and network capabilities
  - Support evolution of a new inter networking architecture
    - Enable: stepwise introduction of new functionality
    - Easy introduction of new functionality without being dependent on agreements with vendors / providers
      - Of course standardization is still required
    - Current network constraints, e.g.
      - Mobile or wired network access
      - A Network may require to use authentication, when prioritization is requested
  - Of course standardization is still required

- Short term flexibility:
  - the capability of a system to adapt itself and react to network conditions and application requirements
  - Dynamic adaption of a new inter networking architecture to:
    - Requirements of current application, e.g.
      - Different behavior for regular or emergency phone calls
    - Current network constraints, e.g.
      - Mobile or wired network access
      - A Network may require to use authentication, when prioritization is requested
    - Capabilities of currently involved nodes
      - Adapt to supported functionality. This is important to utilize new functionality

Service-orientation provides high degree of flexibility
→ apply principles of service-orientation to networks
Service domains distinguish responsibilities for creating, maintaining and providing services

- **Application domain**, represent services of application developers, may be reused by other applications
- **Mediation domain**, map application demands to transport (connectivity) capabilities
  - represent services of network providers (e.g. todays ISPs)
- **Connectivity domain**, represent services related to a specific transport technology (e.g. maintainer of an Ethernet-infrastructure, wireless or dark-fiber)

There is no fixed assignment of functionality to the clouds.
Building Blocks
- Self-contained functionality
- Generic and well-defined interfaces, offering services

Protocol Graphs
- Interaction of BB is defined by a description (not code)
- Descriptions can change easier than code
- Placement of a functionality is not fixed

Framework
- Framework for processes workflows
- Does management

Exchange of Building blocks
long term flexibility
Functional Composition generates descriptions of protocol graphs

- Based on
  - Application requirements
  - Available Building Blocks
  - Network Constraints
  - Administrative Policies

- Time for generation
  - At run time, because then most information is available (dynamic)
  - At design time, to create workflows for bootstrap (static)

Service Selection
- Select a service at runtime

Integration of
- Conventional Protocol Stacks
- Design time approaches
- Run time approaches

Functional Composition & Service Selection

short term flexibility
Summary

what is the right glue?

- Internet as a largely distributed software system
  - Apply software oriented methodologies
  - Changing demands and changing capabilities require a flexible network architecture

- Service oriented Architecture
  - It's an architectural proposal **not an implementation**!
  - Long term flexibility
  - Short term flexibility

- An Internet architecture around services
  - Building blocks
  - Functional composition
  - Communication workflows
  - Move complexity from “implicit code interdependencies” to “explicit service interactions”
Prof. Dr. Paul Mueller
Integrated Communication Systems ICSY
University of Kaiserslautern
Department of Computer Science
P.O. Box 3049
D-67653 Kaiserslautern
Phone:    +49 (0)631 205-2263
Fax:      +49 (0)631 205-30 56
Email:    pmueller@informatik.uni-kl.de
Internet: http://www.icsy.de