Basic Abstractions for an Autonomic Network Architecture

Christophe Jelger, Christian Tschudin, Stefan Schmid, Guy Leduc

UNI BASEL

NEC Empowered by Innovation

ULg UNIVERSITÉ de Liège

AOC Workshop, Helsinki, June 2007

UNI BASEL

ANA Project outlook [www.ana-project.org]

- ANA is funded by the European Union.
  - 4 years: January 2006 to December 2009.
  - 10 European partners, 1 Canadian partner.
  - Initiated by University of Basel, coordinated by ETH Zürich.

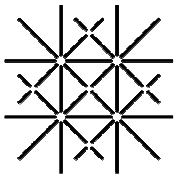
- A Future and Emerging Technologies (FET) project.
  - Forward looking and "risky" research.

- Proactive initiative on Situated and Autonomic Communications (SAC).
  - New paradigms for communication/networking systems.
  - 4 projects: ANA, BIONETS, Haggle, Cascadas.
  - `http://cordis.europa.eu/ist/fet/comms.htm`

Partners

- ETH Zürich (CH)
- University of Basel (CH)
- NEC (DE)
- Lancaster University (UK)
- Fraunhofer Fokus (DE)
- University of Liege (BE)
- University Pierre et Marie Curie (FR)
- NKUA (GR)
- University of Oslo (NO)
- Telekom Austria (AU)
- University of Waterloo (CA)

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich



Fraunhofer Institute for Open Communication Systems

UNIVERSITY OF OSLO

ANA Project outlook

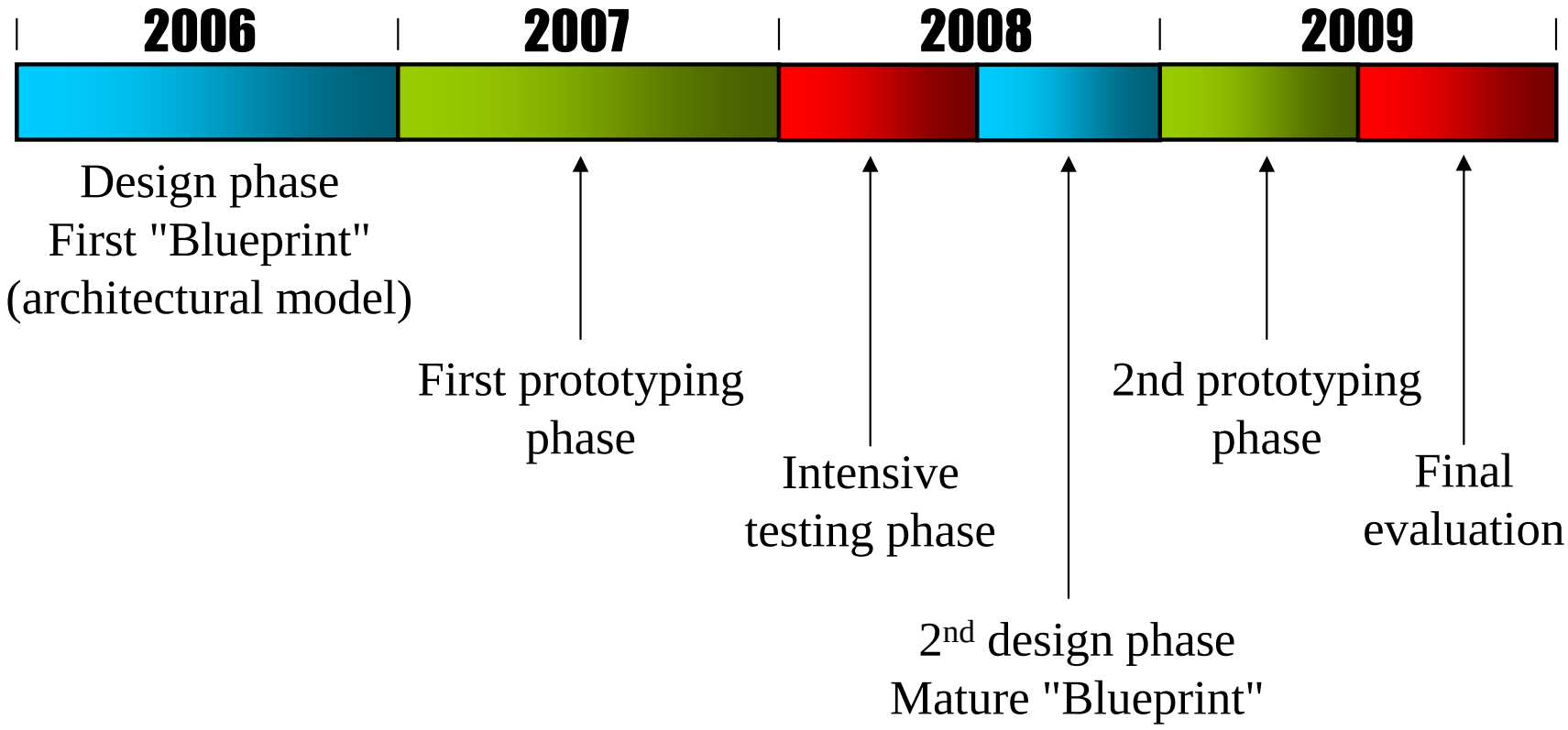
- Goal: To demonstrate the feasibility of autonomic networking.
  - Identify fundamental autonomic networking principles.
  - **Design and build** an autonomic network architecture.

- ANA in a blink:
  - Network must not only **scale** in size but also **in functionality**.
  - **Evolving network**: variability at all levels of the architecture.
  - ANA = framework for **function (re-)composition**.
  - Dynamic **adaptation and re-organization** of network.

- Networks have to work: doing research through prototypes.
  - Early **build** an experimental network architecture.
  - **Prototype used as feedback** to refine architectural models.

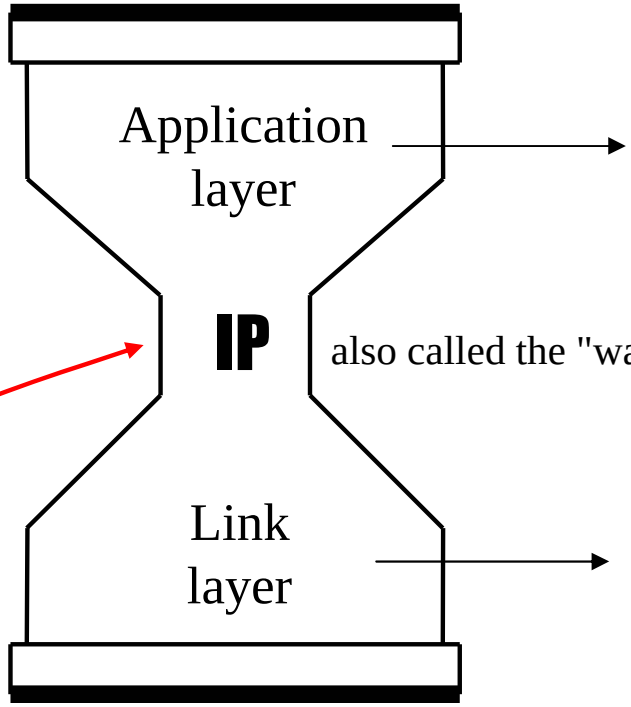
Project organization

- Project is articulated around 2 prototyping cycles.
  - "Learn by doing": design, test/validate, refine.



Motivation

- Variability in the Internet is above and below IP: it's the "hour-glass" model.



www, email, ftp, ssh, DNS,
peer-to-peer (eMule, BitTorrent)
VoIP (Skype), VoD, grid, …

also called the "waist" of the Internet

Ethernet, wifi (802.11), ATM,
SONET/SDH, FrameRelay,
modem, ADSL, Cable, …

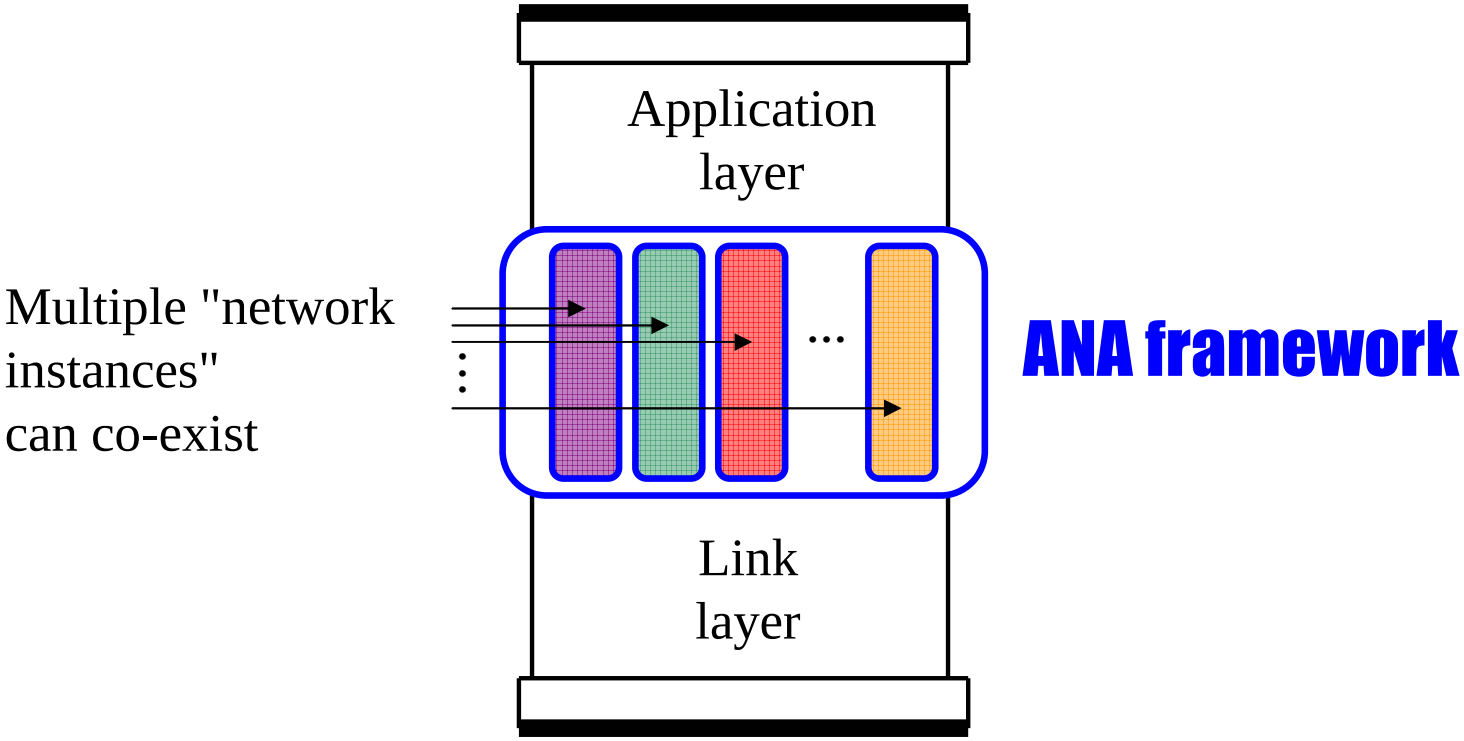
Changing/updating the Internet core (e.g., IPv6, Multicast, MIP, QoS, …) is difficult or impossible !

⇒

Disruptive approach inevitable, this calls for new architecture

ANA ≠ "one-size-fits-all"

- ANA does not want to propose another "one-size-fits-all network waist".
  - ANA is a **meta-architecture** to host, interconnect, and federate multiple heterogeneous networks.



ANA: a least common denominator …

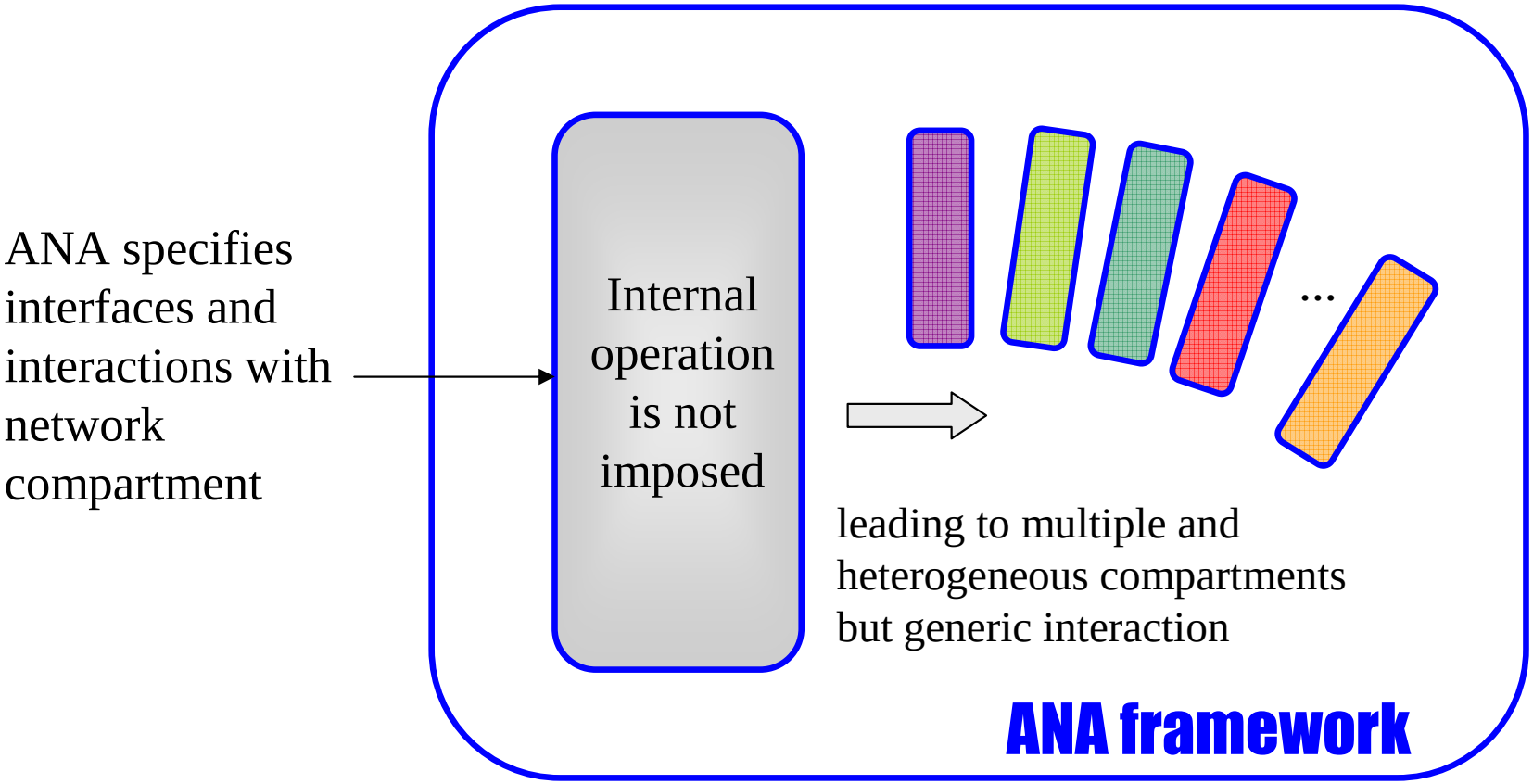
- Pitfalls in network architecture design:
  - Static/rigid standards instead of mechanisms for change management
  - Built-in address dependency (i.e. address-centric architecture)
  - Global address space (requires uniqueness and global coordination)
  - Leaking of and relying on network internal details

- ANA abstractions (to be detailed in the following slides):
  - Compartment.
  - Information Channel (IC).
  - Information Dispatch Point (IDP).
  - Functional Block (FB).

- OSI next generation?... kind of, but with clear APIs and mechanisms.

ANA compartment

- Compartment = wrapper abstraction for networks
- ANA does not impose how network compartments should work internally: the ANA framework specifies how networks interact.



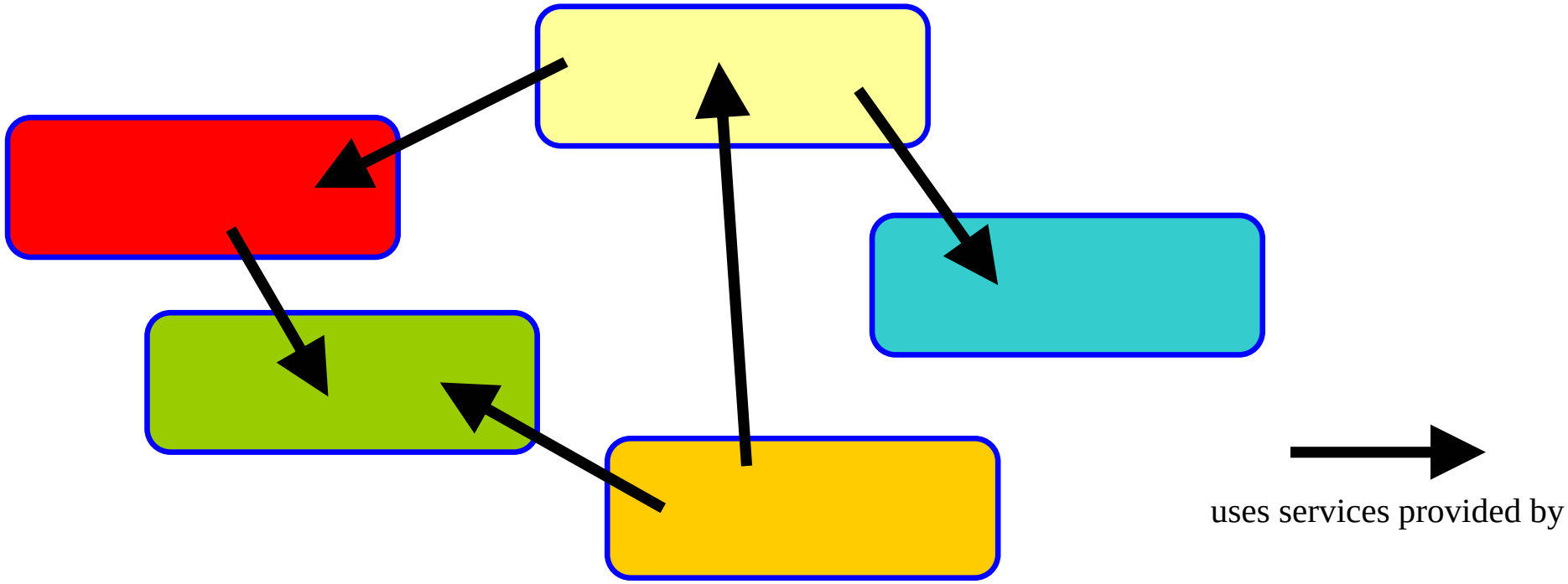
Network compartments

- A (network) compartment implements the operational rules and administrative policies for a given communication context. It defines:
  - How to join and leave a compartment: **member** registration, trust model, authentication, etc.
  - **How to reach** (communicate with) another member: peer resolution, addressing, routing, forwarding, etc.
  - The compartment-wide policies: **interaction rules** with "external world", the **compartment boundaries** (administrative or technical), peerings with other compartments, etc.

Compartments decompose communication systems and networks into smaller and easier manageable units.

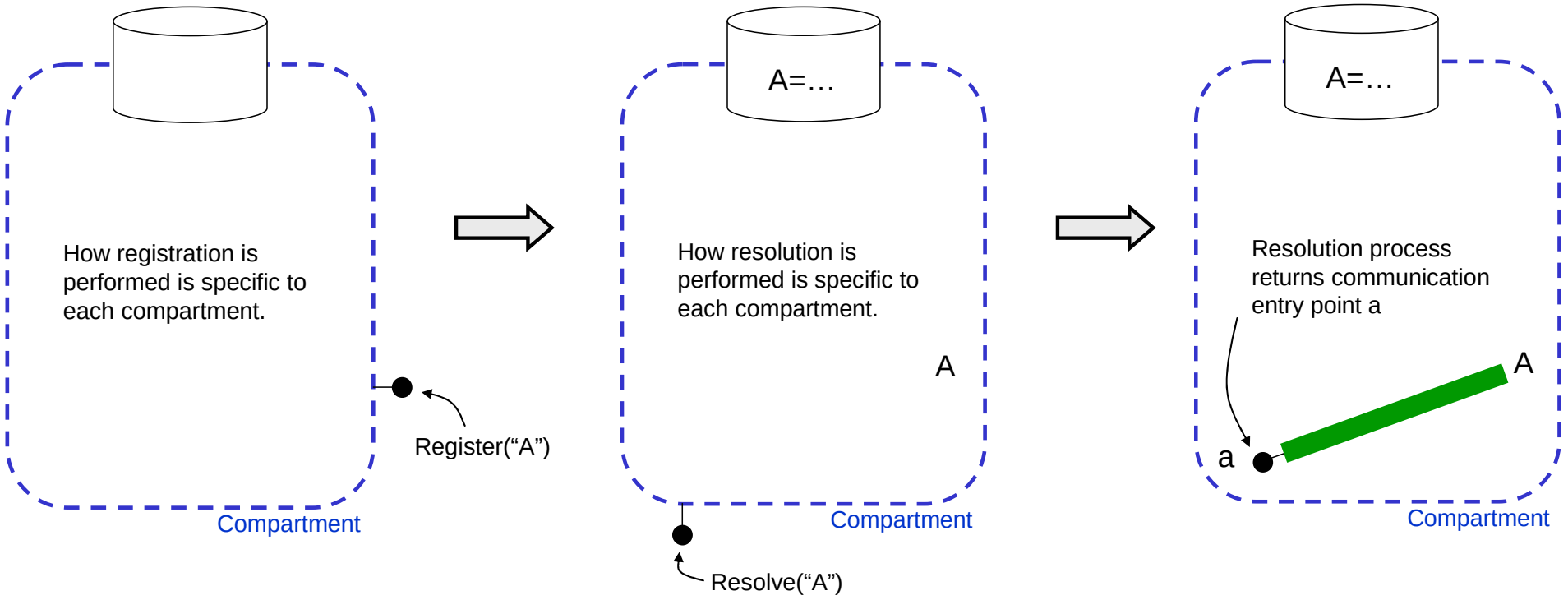
Network compartments (cont.)

- In addition, the compartment abstraction serves as the unit for the federation of networks into global-scale communication systems.
  - Compartments can be overlaid, i.e. compartments can use the communication services of other compartments (and vice versa).
    - Similar to OSI user-provider concept with generic SAP primitives.



Network compartments (cont.)

- **Registration** and **resolution** are key functionalities of compartments.
  - Conceptually, each compartment maintains a membership database.
  - Registration: explicit membership is required ("default-off" model).
  - Resolution: explicit request before sending ("no sending in the void").



What about addresses and names?

- Addressing and naming are left to compartments.
  - Each compartment is free to use any addressing and naming schemes (or is free to not use addresses, for example in sensor networks).

- The main advantages are:
  - No need to manage a unique global addressing scheme.
  - No need to impose a unique way to resolve names.
  - ANA is open to future addressing and naming schemes.

- The main drawbacks are:
  - Global routing becomes something similar to searching. (if communicating parties are not all members of a given compartment).

What about addresses and names? (cont.)

- Main challenges for ANA are:
  - How to **gateway heterogeneous** compartments.
  - How to **federate and overlay** compartments.
  - How to access compartment services in a **generic manner**.
  - How to **recursively resolve** a name through the collection of compartments.

- How does ANA support heterogeneous addressing and naming schemes?
  - "Clients" of ANA interact with generic primitives and objects.
  - Packet dispatch in the ANA core is based on "network pointers".
  - Access to compartments is via generic functions.

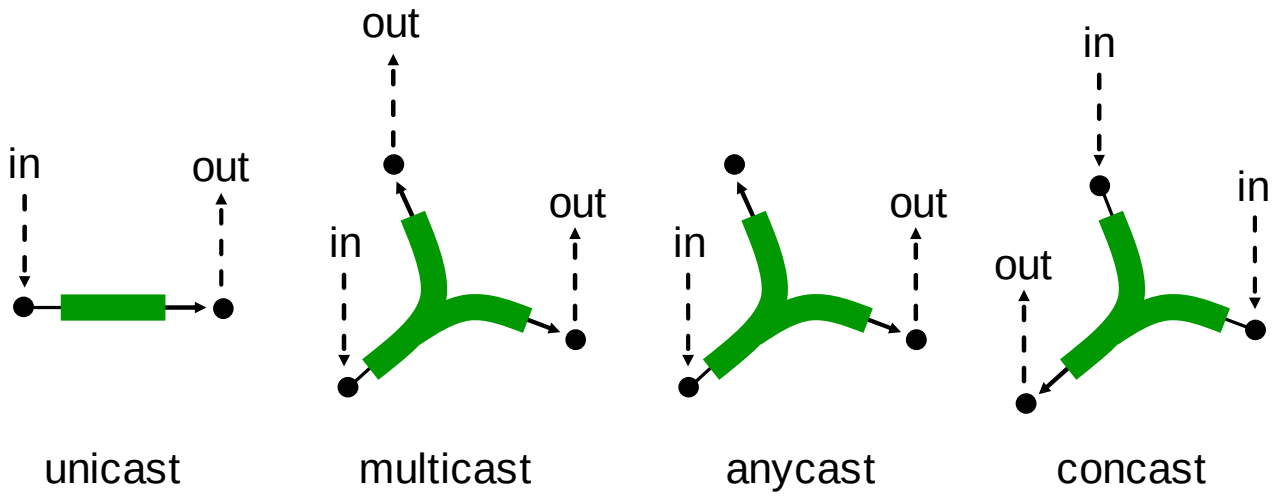
Key concept is to cleanly separate compartment internals from abstractions exposed to users of compartment services.

Information channels (ICs)

- Resolution process returns access to an "information channel" that can be used to reach the target member(s).

  (Something like an OSI (N)-connection)

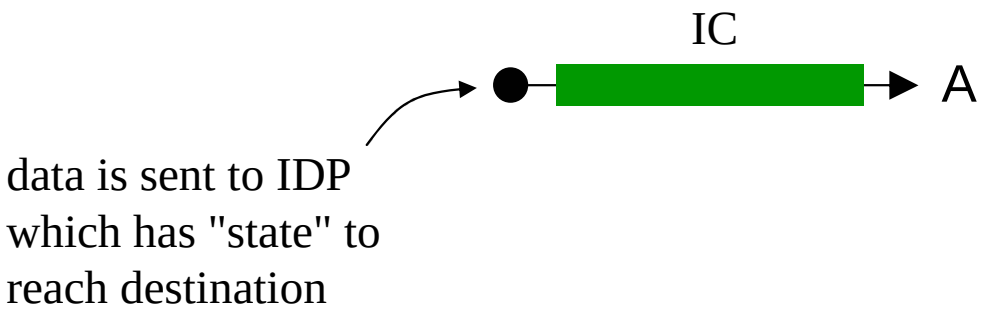
Various types of information channels.



Information dispatch points (IDPs)

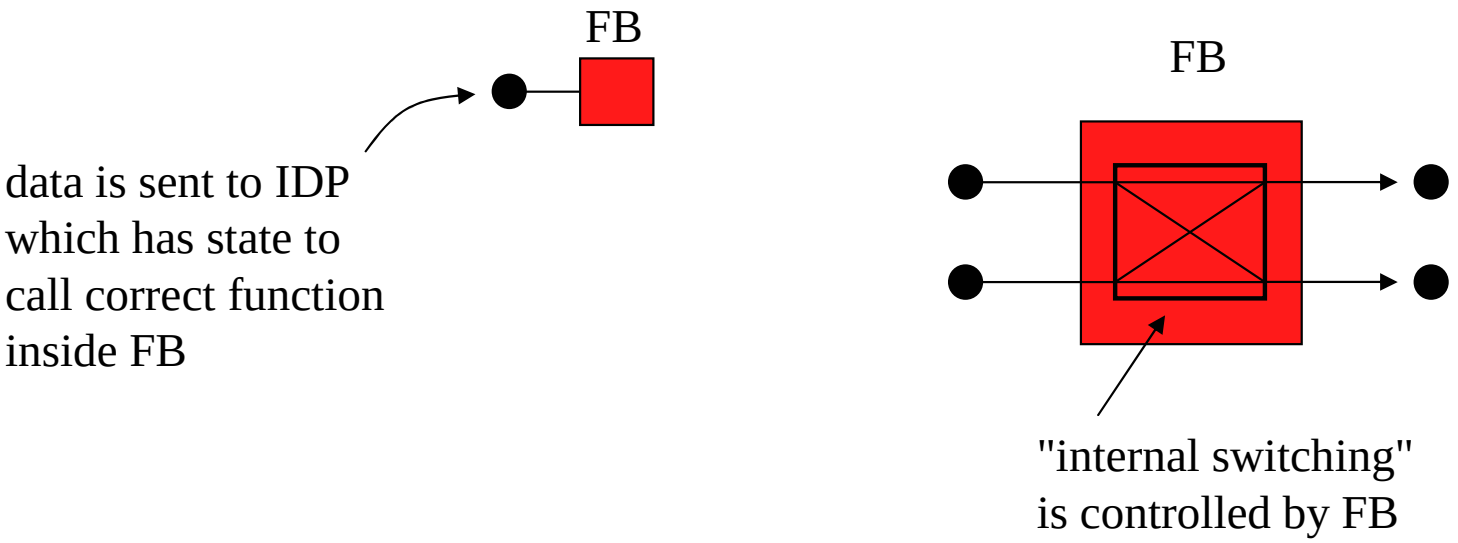
- **Startpoints** instead of endpoints

  - In ANA communication is always towards a startpoint, or information dispatch point (IDP) (akin to an OSI SAP+CEP).
    - Ability to bind to destinations in an address agnostic way.
    - This is important to support many flavors of compartments that can use different types of addresses and names.
    - Useful decoupling between identifiers and means to address them.

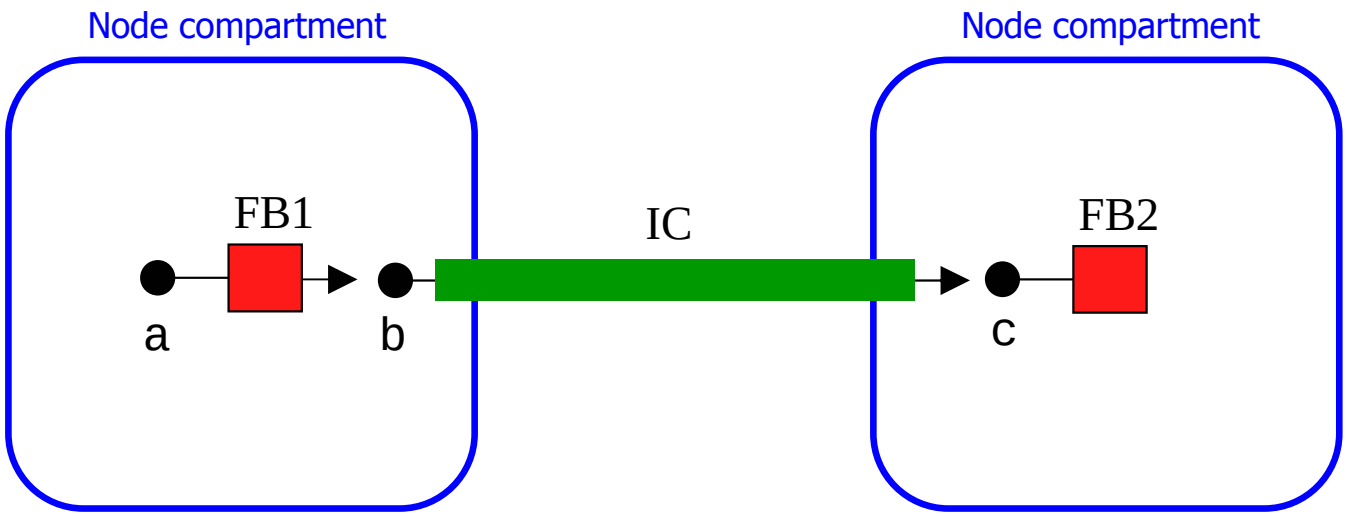


Functional blocks (FBs)

- Code and state that can process data packets.
  - Protocols and algorithms are represented as FBs.
  - Access to FBs is also via information dispatch points (IDPs).
  - FBs can have multiple input and output IDPs.
  - FB internally selects output IDP(s) to which data is sent.



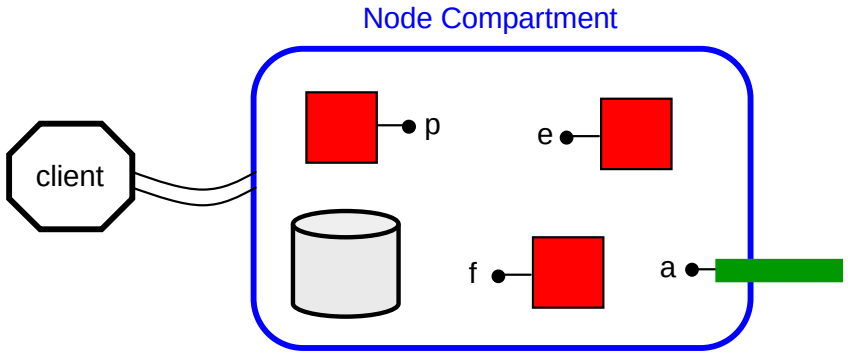
How ICs, FBs, and IDPs fit together



Modeling nodes as compartments

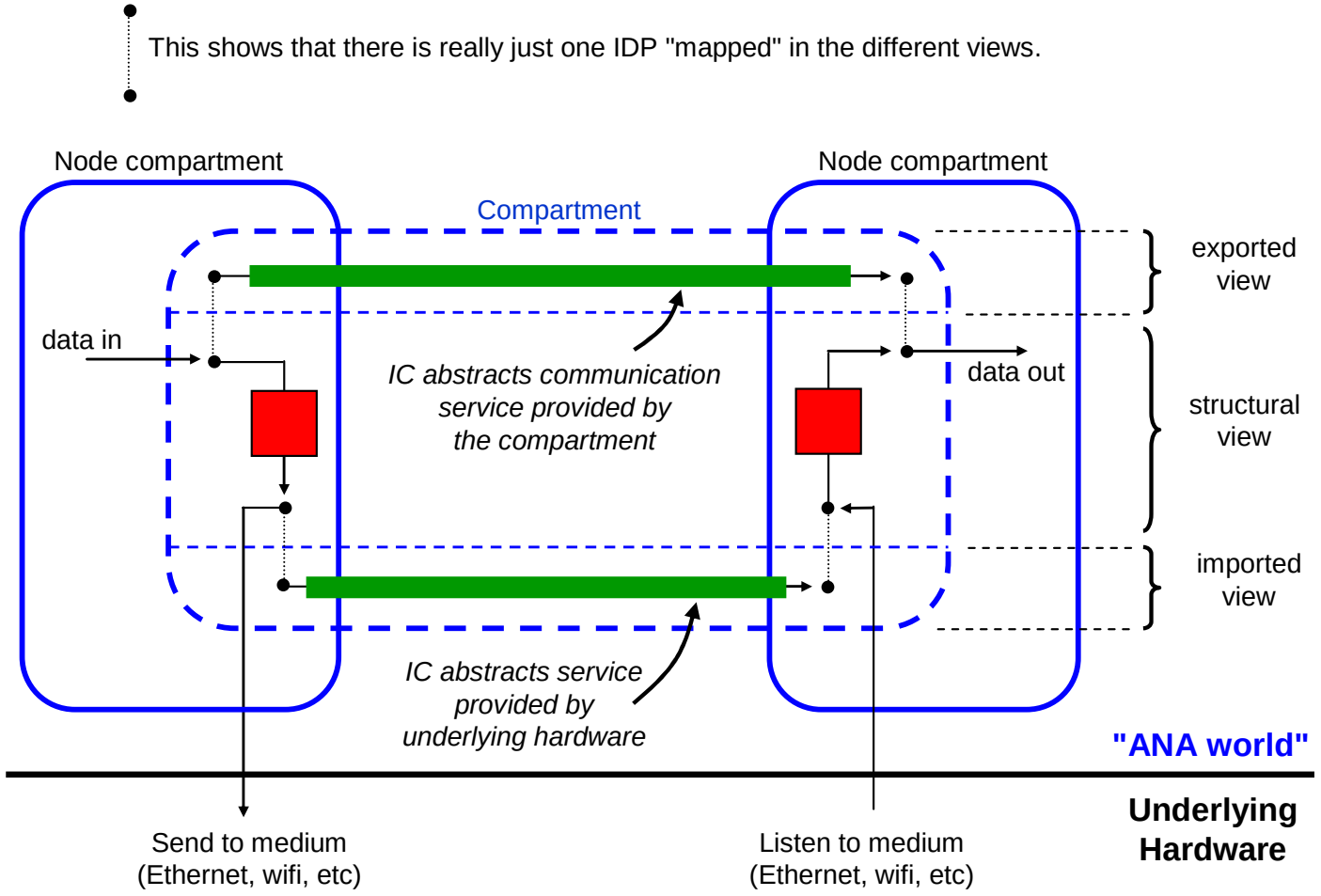
- Organize a node's functionalities as (compartment) members:
  - Member database: catalog of available functions (FBs).
  - Resolution step to access a given function.
    - Also implements access control.
  - Resolution instantiates functional blocks (FBs).
  - The node compartment hosts/executes FBs and IDPs.

- Applications first attach to the node compartment:
  The node compartment is the "startpoint" of any communication.



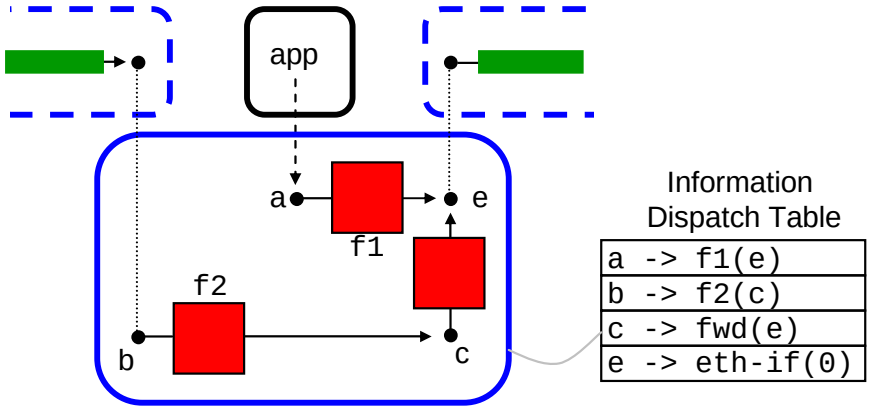
Different "views" for a compartment

- A network compartment has different views, for different usage.



Functional composition

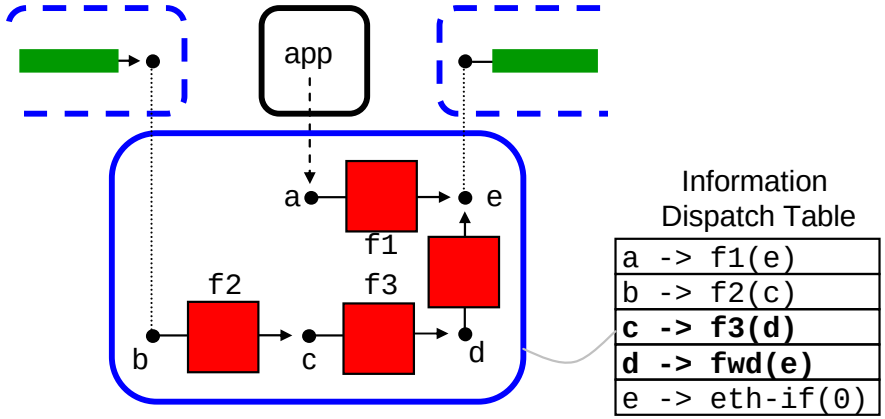
- "Chains" of functions are setup on-demand in a dynamic way.
  - Packet dispatching in ANA is based on IDPs.



Information Dispatch Table

a -> f1(e)
b -> f2(c)
c -> fwd(e)
e -> eth-if(0)

re-binding of IDP 'c' is not visible to users of 'c' (function `f2` here)

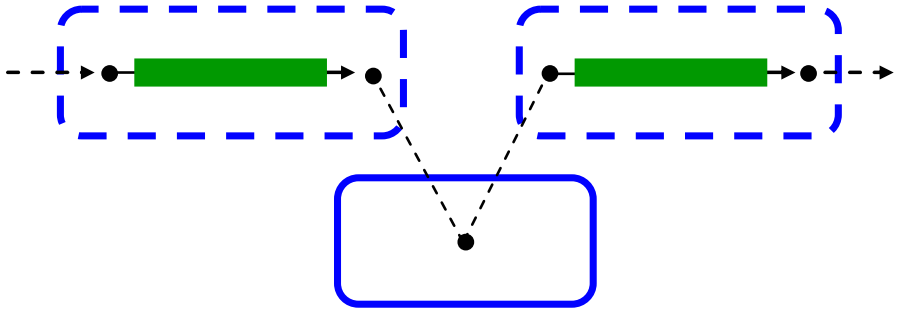


Information Dispatch Table

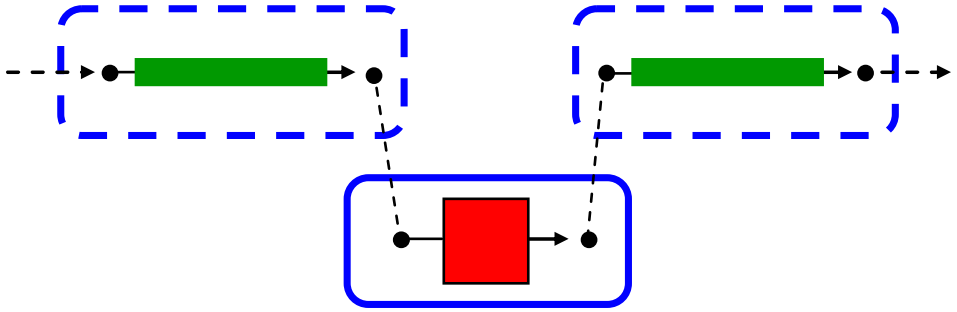
a -> f1(e)
b -> f2(c)
c -> f3(d)
d -> fwd(e)
e -> eth-if(0)

re-binding is a simple change in dispatch table

Forwarding … some examples.

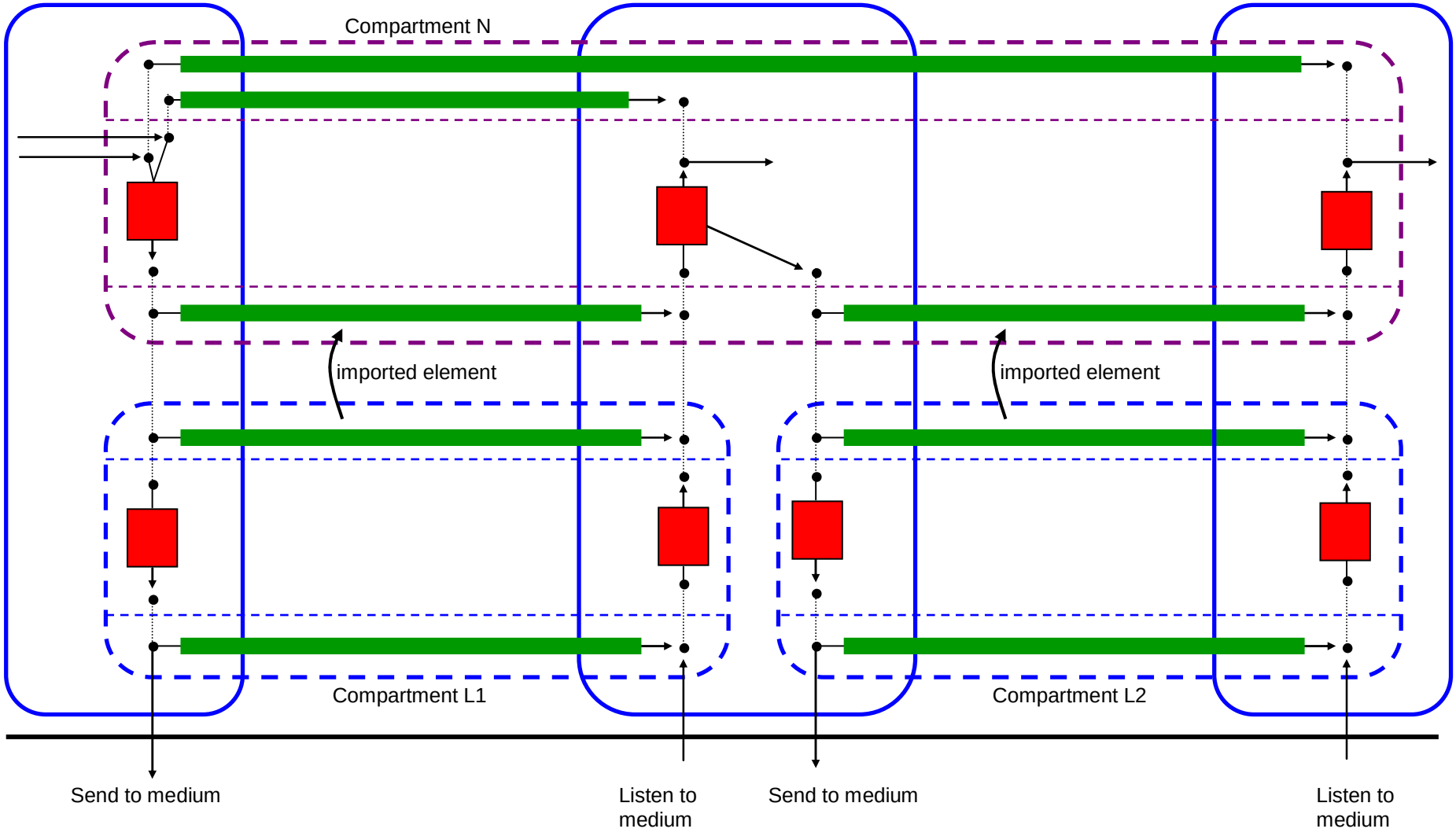


Bridging



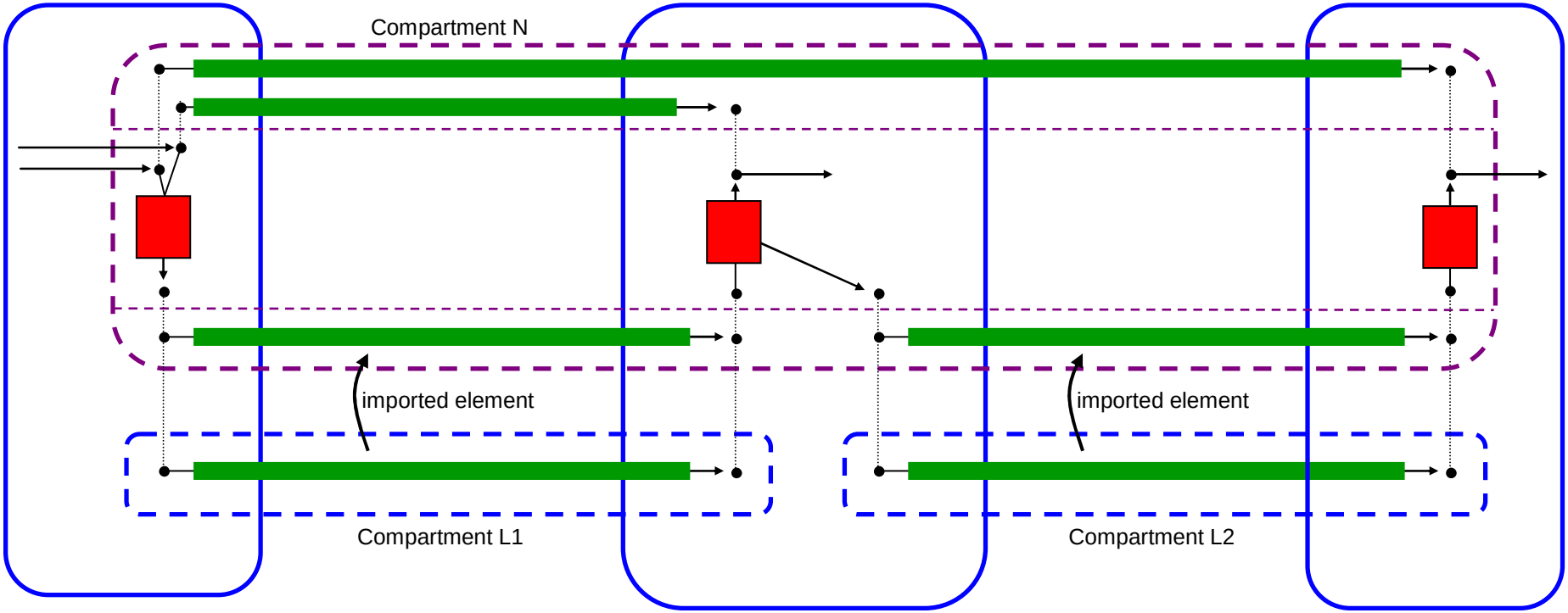
+ intermediate processing

Overlay scenario with compartments



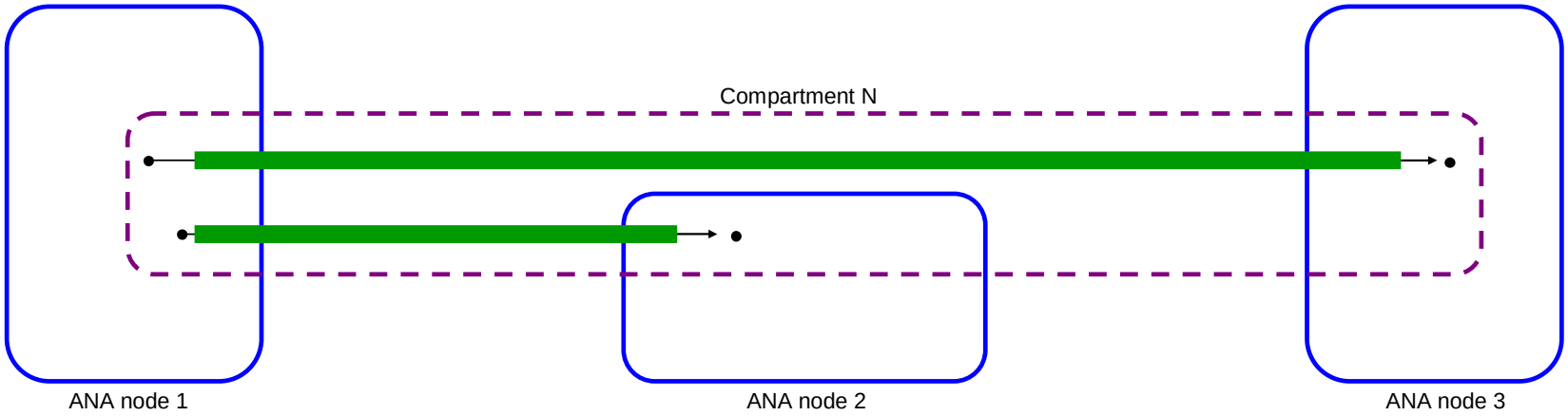
Overlay scenario with compartments (2)

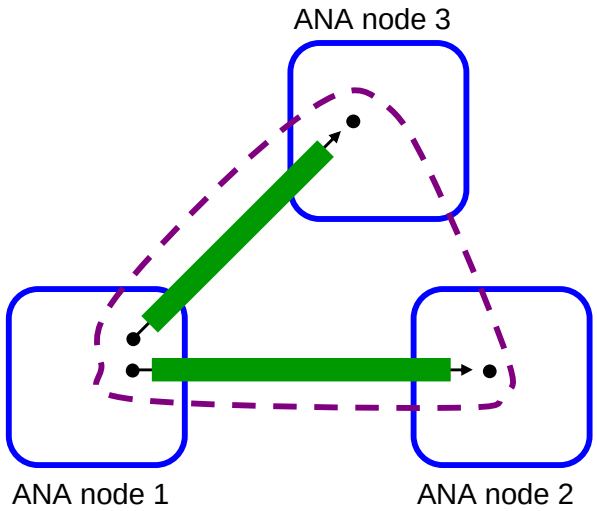
Same figure but only with exported views of L* compartments.



Overlay scenario with compartments (3)

Figure just showing export view of compartment N.





Which could also be drawn like that (just showing the export view).

Where does autonomic fit into the Blueprint?

- ANA Blueprint offers a flexible and evolvable framework.
  - Allows variability at all levels of the architecture: multiple
    - **functionalities,**
    - **variants to perform a given task,**
    - **and compartments**

    co-exist and (can) compete, open for extensions (evolution).

- Blueprint provides a **well-defined structure** on which to operate in an autonomic way.
  - Easy to test/replace/upgrade parts of the system (except for minimal core).
  - Generic set of abstractions provides "common language" for algorithms and protocols.

Prototyping in ANA

- Prototyping is an essential activity in ANA.
  - After 1st prototyping phase, architecture will be refined (starting July 2008).
  - Feedback from testing phase is critical.

- Project is currently developing an ANA node prototype.
  - All functionalities of ANA + an abstraction layer to run ANA on different operating systems or on dedicated hardware (e.g. network processors).

- To increase size of testbed we plan to:
  - Allow multiple ANA nodes to run on the same physical device.
    - i.e. With link emulation by abstraction layer.
  - Develop an initiative to attract external researchers: **ANA@Home**
    - To build a large-scale and open research platform for autonomic networking.

Prototyping in ANA (cont.)

- To manage the testbed we want to develop a distributed management system.
  - To see in real-time active ANA nodes and links.
  - To add/remove experimentation links on-demand or via scheduling system.
  - To manage ANA nodes (e.g. software updates, reboot, etc).

- **ANA@Home**
  - Objective is to attract external participants to use the ANA testbed.
  - Increase size and relevance of testbed.
  - Increase "processing power" and heterogeneity.
  - Encourage external developments inside ANA framework.
  - In practice ANA@Home = ANA node accessing ANA testbed via dedicated points of presence (POPs).

Next steps in ANA

- In 2007 …
  - Initial deployment of testbed (PCs, wireless access points, switches).
  - Development of testbed management system/console.
  - Development of ANA node prototype.
  - Development of ANA@Home initiative.
  - Development of network compartments modules (FBs).

- Objective are:
  - To demonstrate self-configuration and self-organization of network nodes inside network compartments.
  - To demonstrate autonomic behavior of e.g. compartment discovery and path setup, resilience, network monitoring.

Thank you for your attention

Questions?

To learn more, google "ANA project"