# Information Dispatch Points

Christophe Jelger
**University of Basel, Switzerland**
christophe.jelger@unibas.ch

NetArch Symposium
Ascona, Switzerland, 17th March 2009

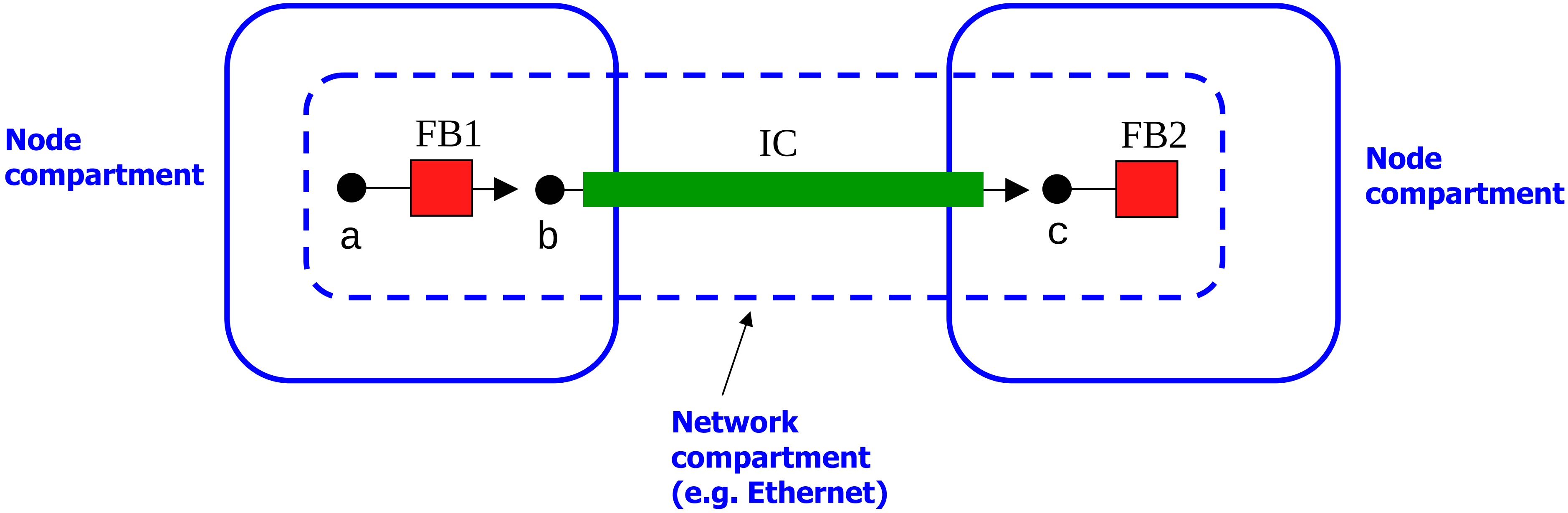
# Research Context

- EU funded ANA Project (FP6, FET, SAC)
  - 4 years (2006-2009)
  - 10 European partners, 1 Canadian partner

- Goal of the project: demonstrate self-* properties with a running prototype.
  - On the network architecture side: build a minimal network node/system ready to host new (and disruptive) protocols and networking schemes.

# ANA in a blink: abstractions

- **Compartment**: "wrapper" for networks
- **Information Channel** (IC): generic communication channel.
- **Information Dispatch Point** (IDP): generic indirection system.
- **Functional Block** (FB): packet processing entity.



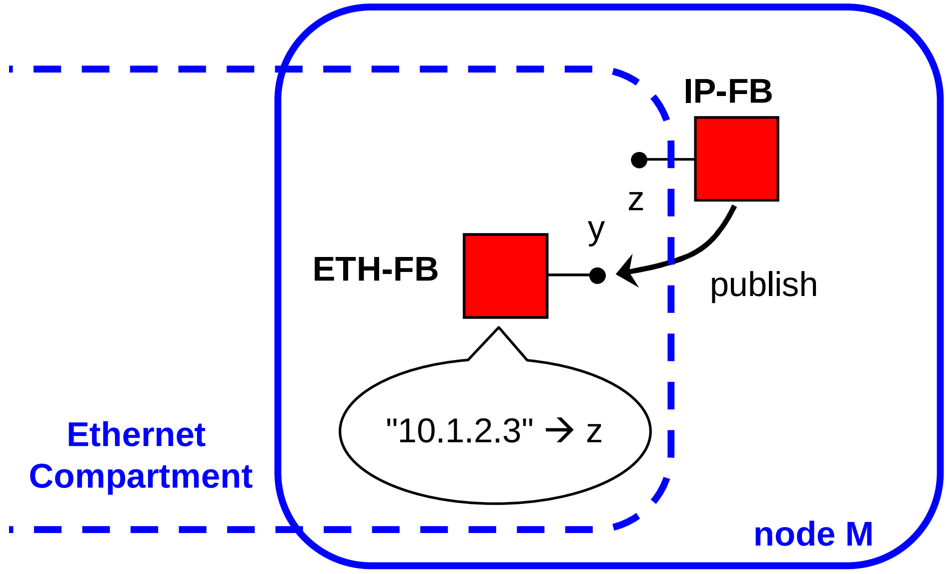
# ANA in a blink: API

The API has 6 fundamental primitives.

- IDP$_p$ publish(IDP$_c$, CONTEXT, SERVICE)
- int unpublish(IDP$_c$, IDP$_p$)
- IDP$_r$ resolve(IDP$_c$, CONTEXT, SERVICE)
- int release(IDP$_c$, IDP$_r$)
- void* lookup(IDP$_c$, CONTEXT, SERVICE)
- int send(IDP$_r$, DATA)

# ANA in a blink: example

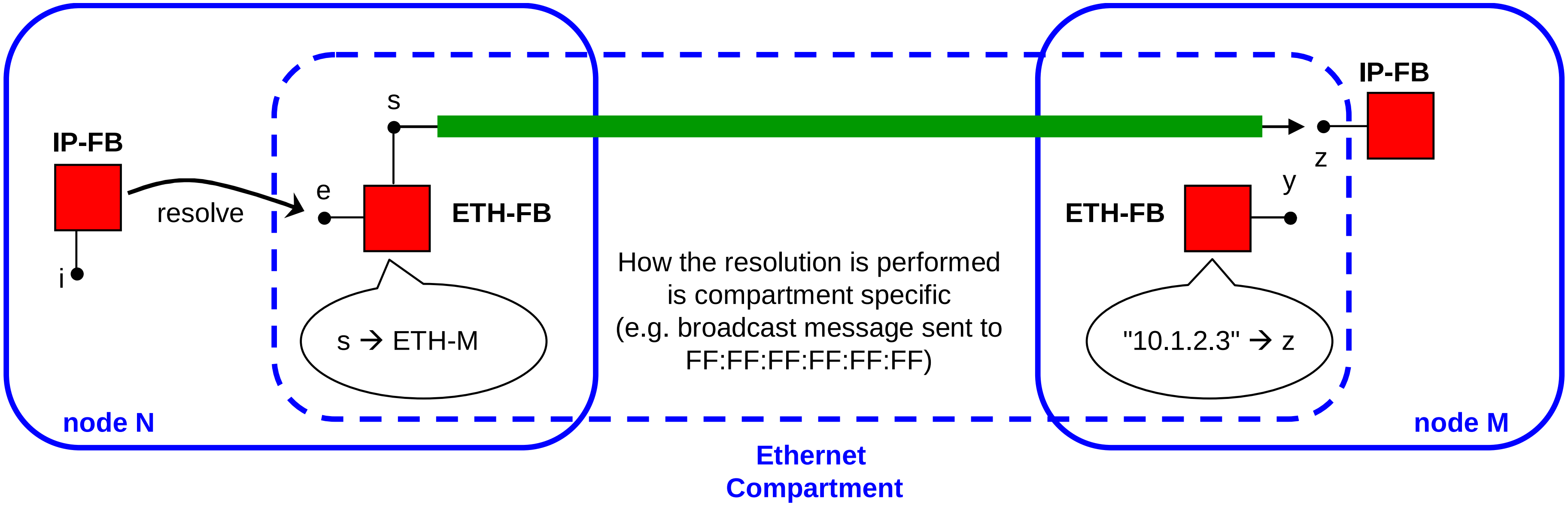
Publishing an IPv4 address in the Ethernet compartment.



z <-- publish(y, "*", "10.1.2.3)

# ANA in a blink: example

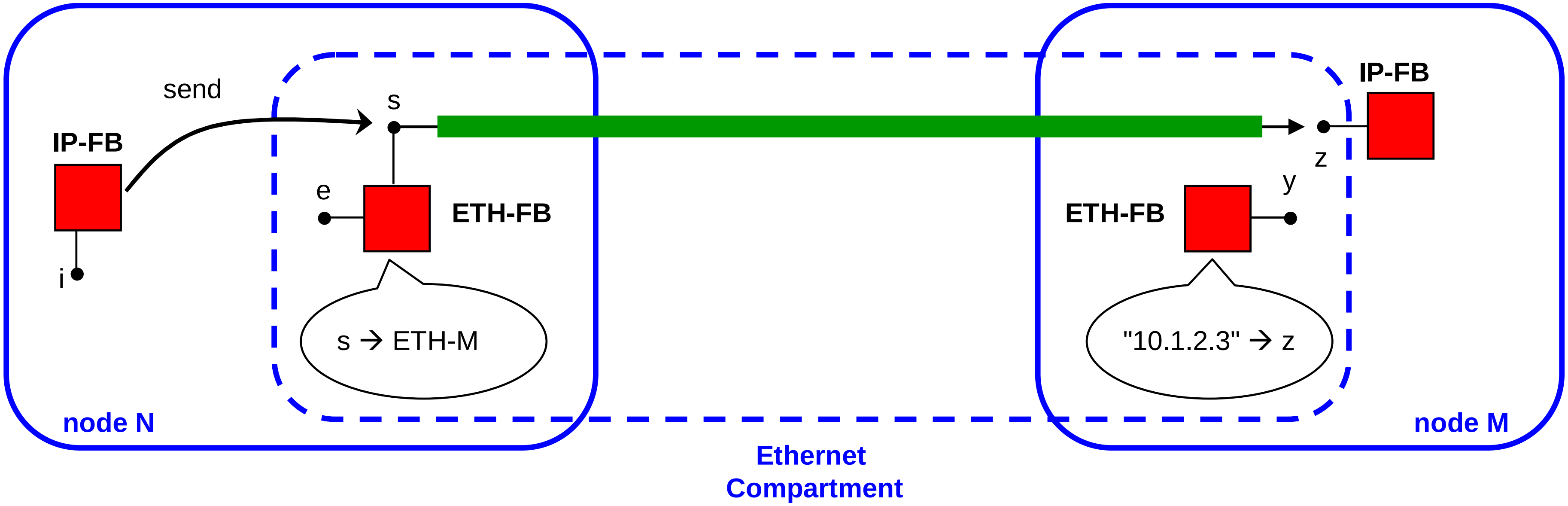
Resolving an IPv4 address in the Ethernet compartment.



s <-- resolve(e, "*", "10.1.2.3")

# ANA in a blink: example

Sending data.



send(s, DATA)

# So what about IDPs?

- Actually the "ANA Core" machinery solely operates on IDPs:
  - – packet forwarding inside the node is based on IDPs.
    - Each IDP has a unique node-local flat value.
  - – all components inside the node are identified by IDPs.
  - – all bindings between components are via IDPs (and binding information is stored in IDPs)

# So what about IDPs?

- IDPs are more than just indirection points.
  - Each IDP stores a lot of information: binding info, information channel info, MTU, access rights, status (ready,busy), + whatever state the component bound to it wants to store (e.g. TCP session info).
  - IDPs can be manipulated dynamically: binding, forking (T shape), redirection.
    - e.g. it's possible to replace a component with a new or optimized version and re-attach all IDPs to the new part.

# Main benefits of IDPs

- The "ANA Core" is simple and extensible: "everything is an IDP".

- Indirection (i.e. flexibility, loose binding) is built-in by default in the core node machinery.

- The "ANA Core" machinery is fully address and name agnostic: it only handles local labels.

# Can I use that to build networks?

- Yes of course: check the ANA website for our Linux-based prototype.
  - Ethernet, IP, + many research-oriented protocols, all developed around the minimum "ANA Core".
- Coming in 2009:
  - A scripting language to dynamically configure the components and run-time behaviour.
    (most likely C-LUA binding with concurrency)

# Thank you for your attention

Questions?

Want to learn more?
Google "ANA Project"