Towards evolving specs of security protocols

Dusko Pavlovic
Kestrel Institute

(based on joint work with A. Datta, J. Mitchell, D. Smith…)

March 7, 2002

Claim

Security Engineering

is a part of

Software Engineering

Claim

- it is helpful to analyze:
  - protocols in context of architectures
  - security as a part of of high assurance
  - malicious attackers on connectors together with unspecified environments of components
- both SE and SE are concerned with
  - distributed,
  - multi-layered,
  - heterogenous complex systems…

Outline

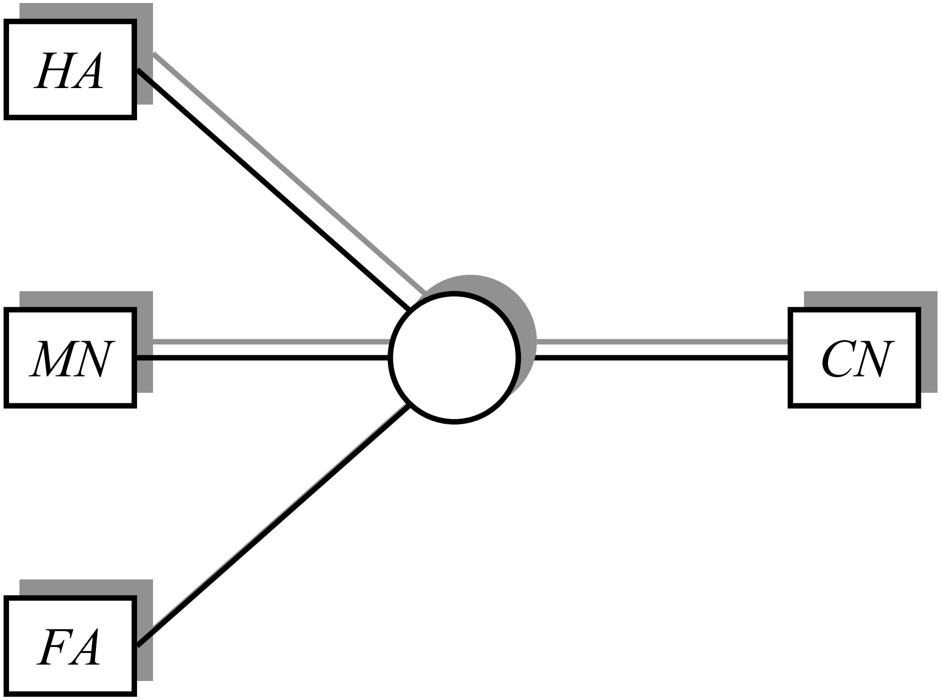
- Mobile proposals:
  - IPv4 *vs* IPv6
- Problem:
  - remote redirection (traffic hijacking)
- Adding authentication:
  - espec transformation
- Variations and ongoing work

Papers

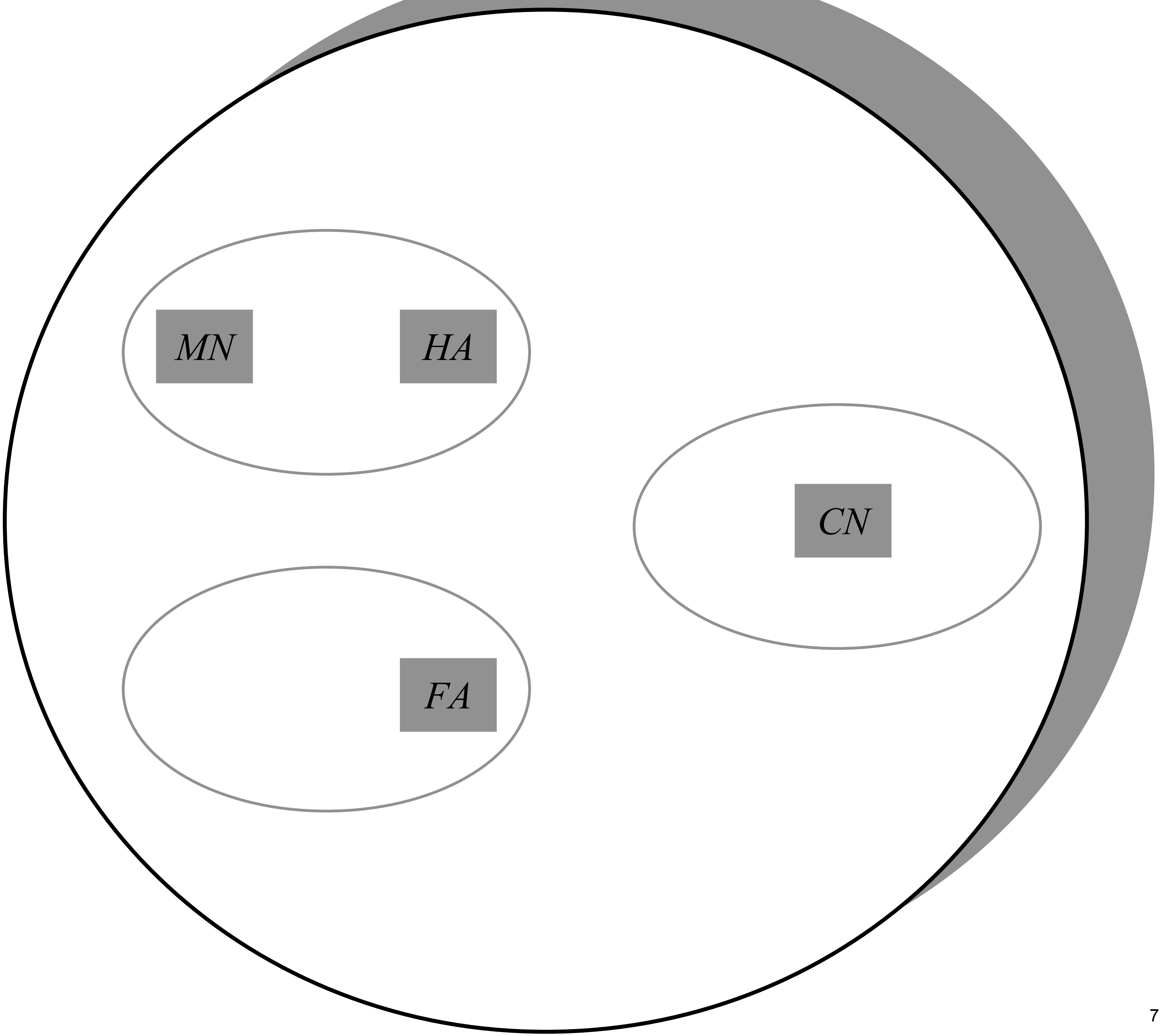
- Authentication for Mobile IPv6
  - with A. Datta, J. Mitchell and F. Muller
- Composition and refinement of behavioral specifications
  - with D. Smith
- Guarded transitions in evolving specifications
  - with D. Smith

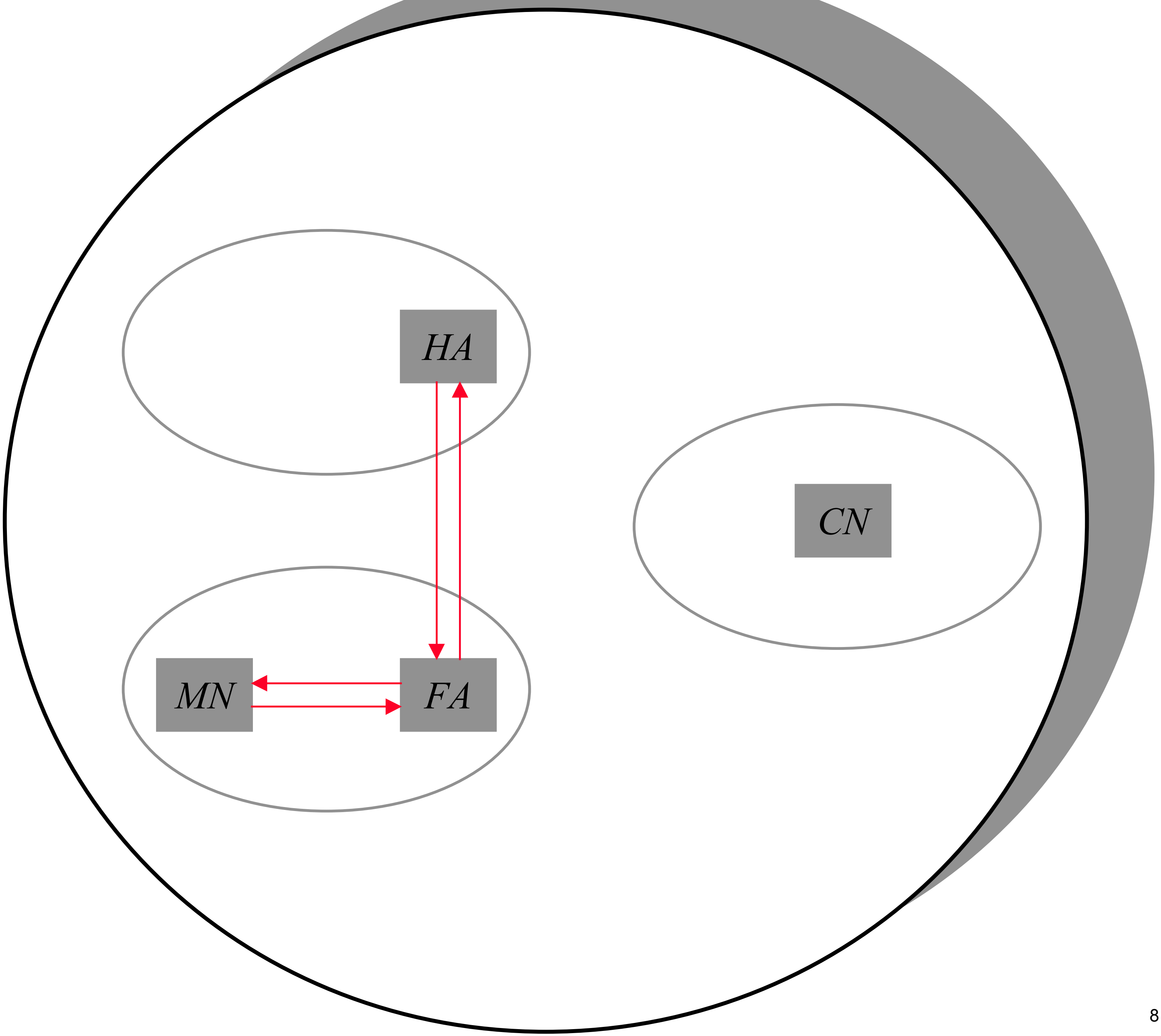
http://www.kestrel.edu/users/pavlovic/

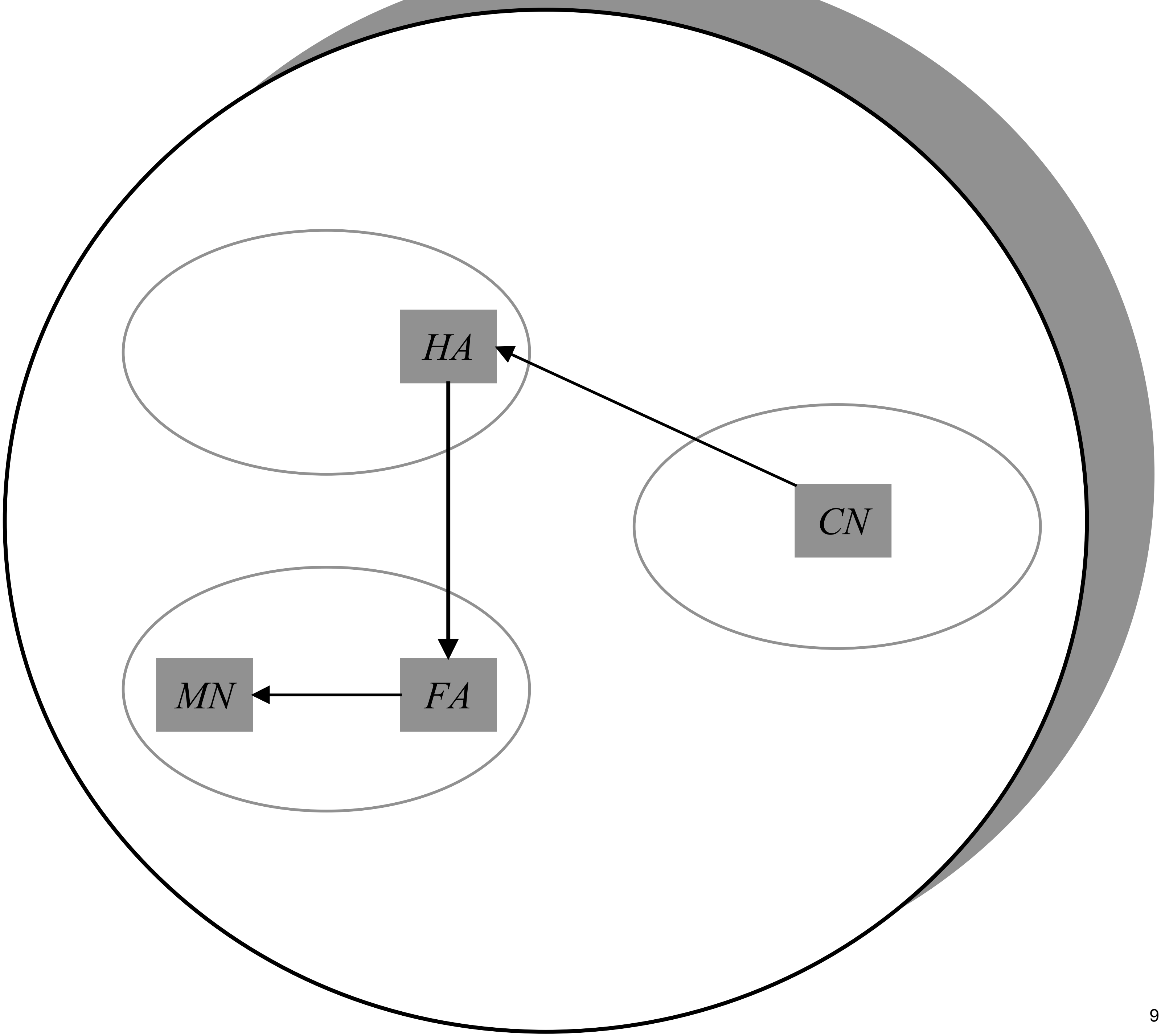
Mobile IPv4

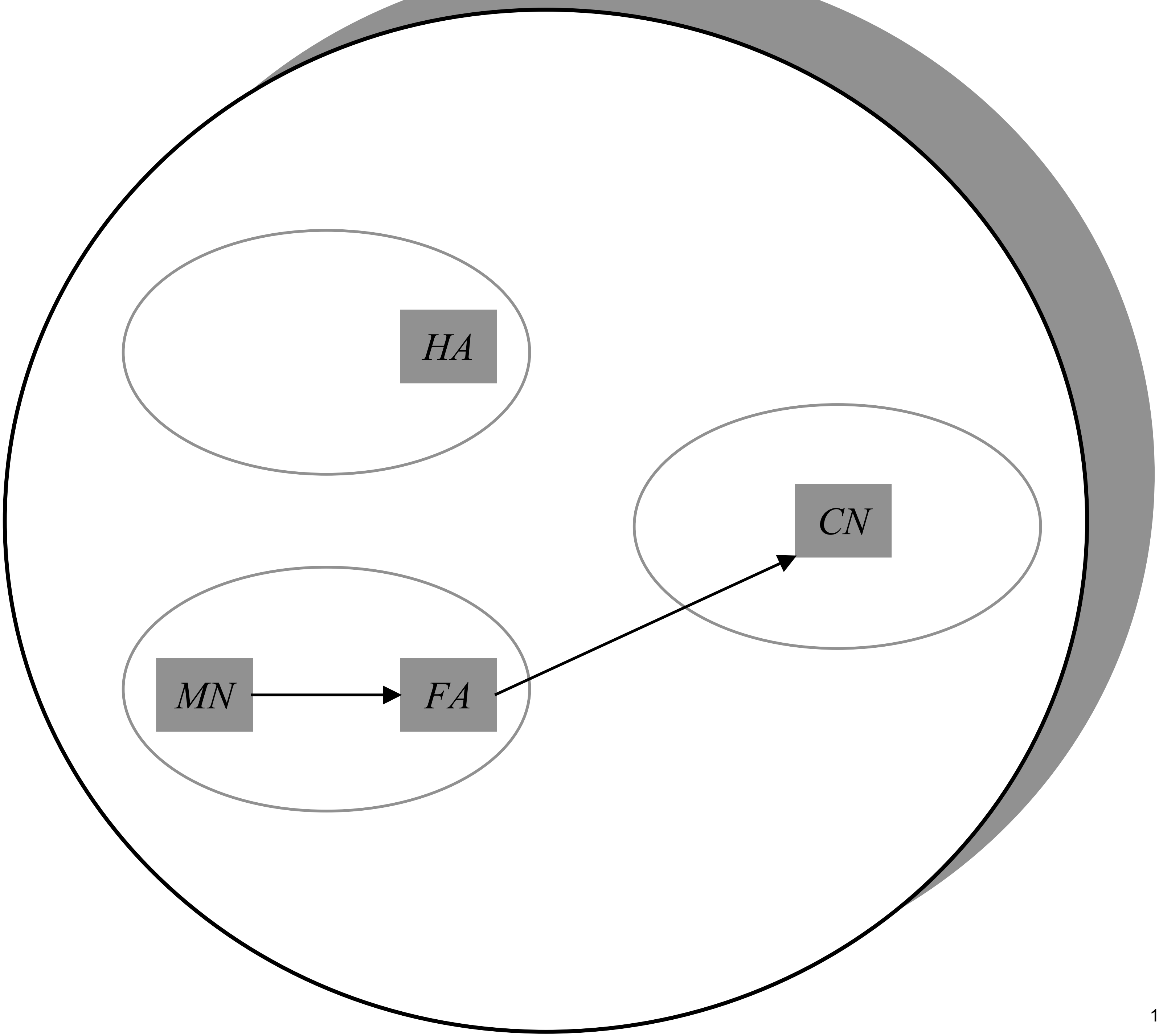


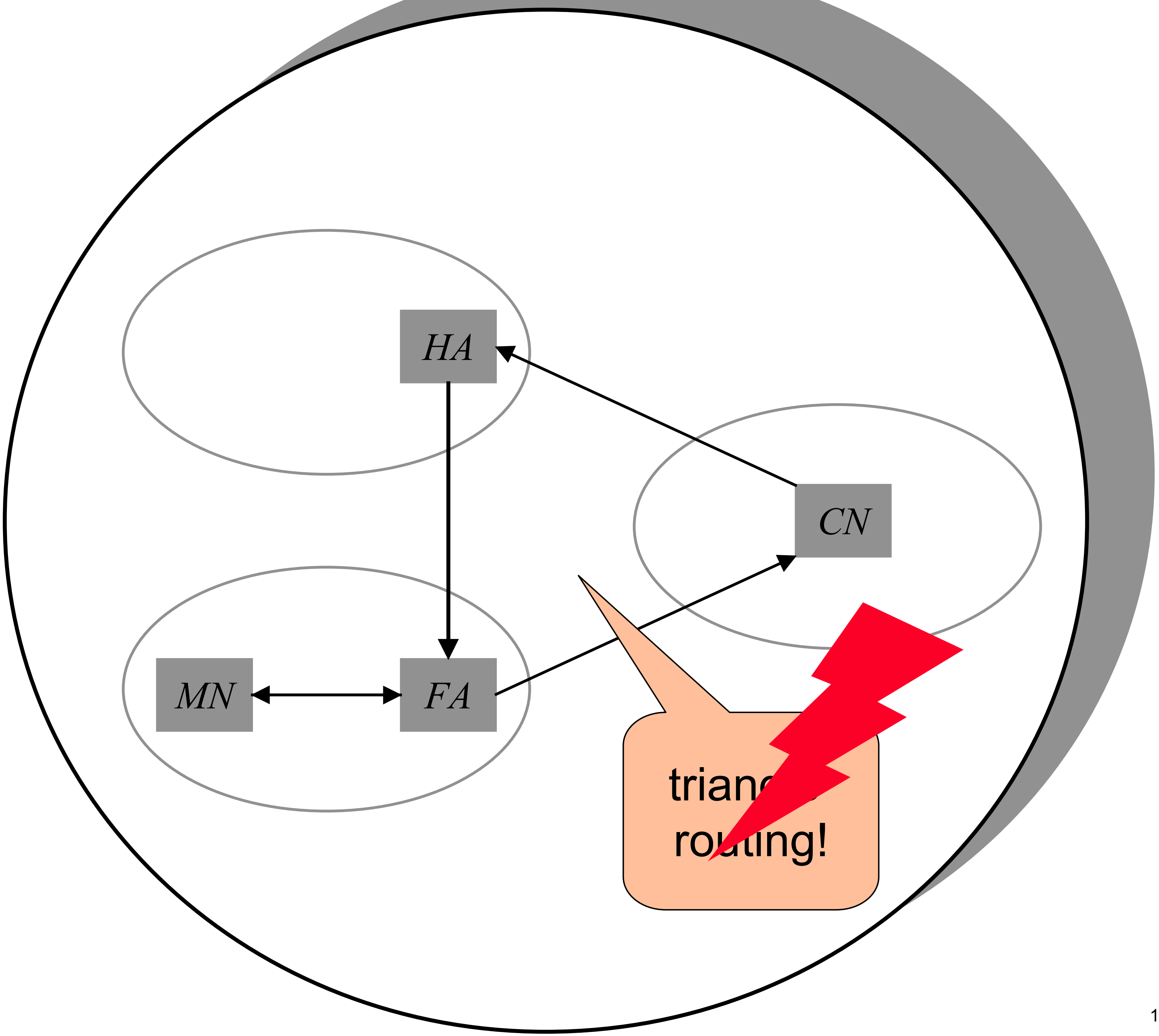
initial architecture


MN
HA
CN
FA

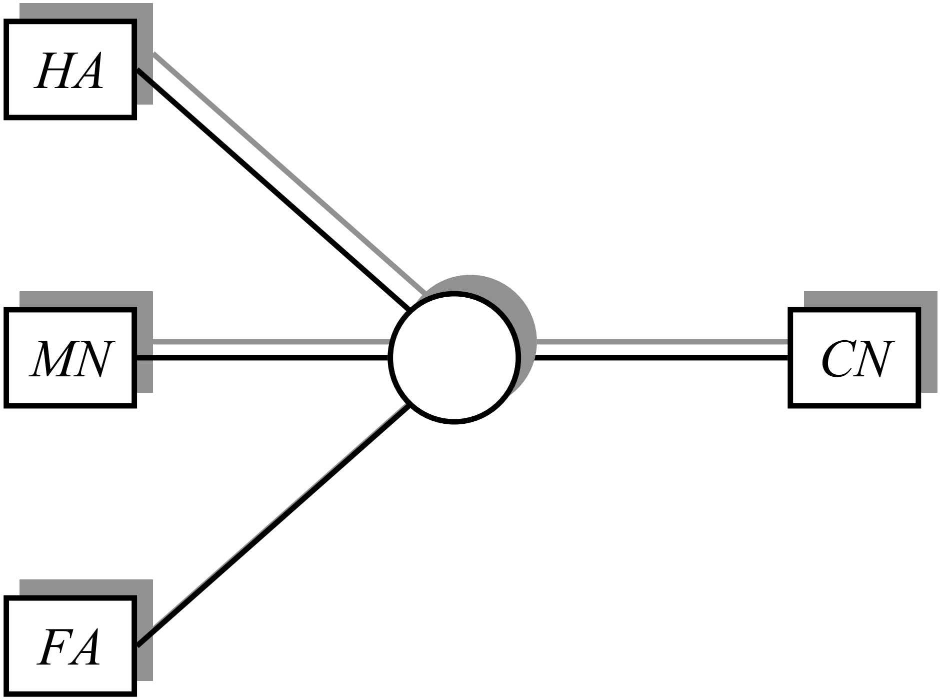

HA
CN
MN
FA


HA
CN
MN
FA


HA
CN
MN
FA


HA
CN
MN
FA
triangle
routing!

Mobile IPv4



session architecture

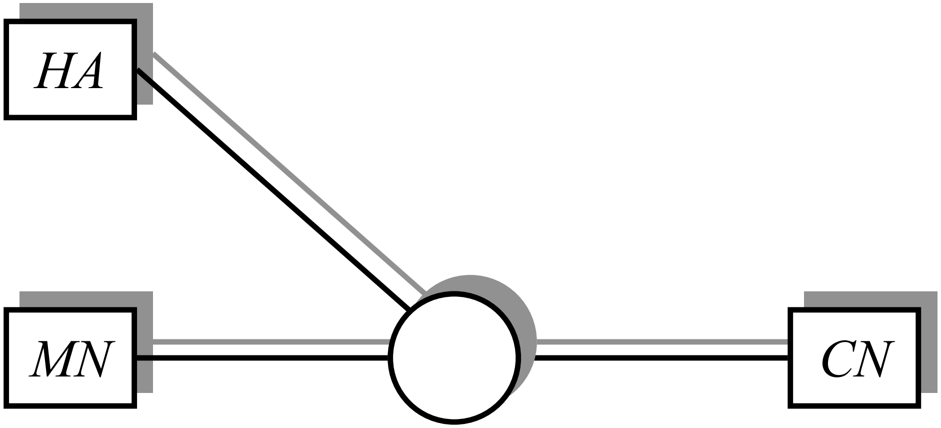
Mobile IPv6

- avoid triangle routing:
  - use IPv6 Routing Header and tunneling
- minimize
  - network partitioning
  - computational load on:
    - routers
    - nodes: no expensive encryptions or decryptions
  - number of messages
  - need for infrastructure: no global PKI
- maximize
  - performance and availability: no DoS
  - end-to-end security: authenticate location information

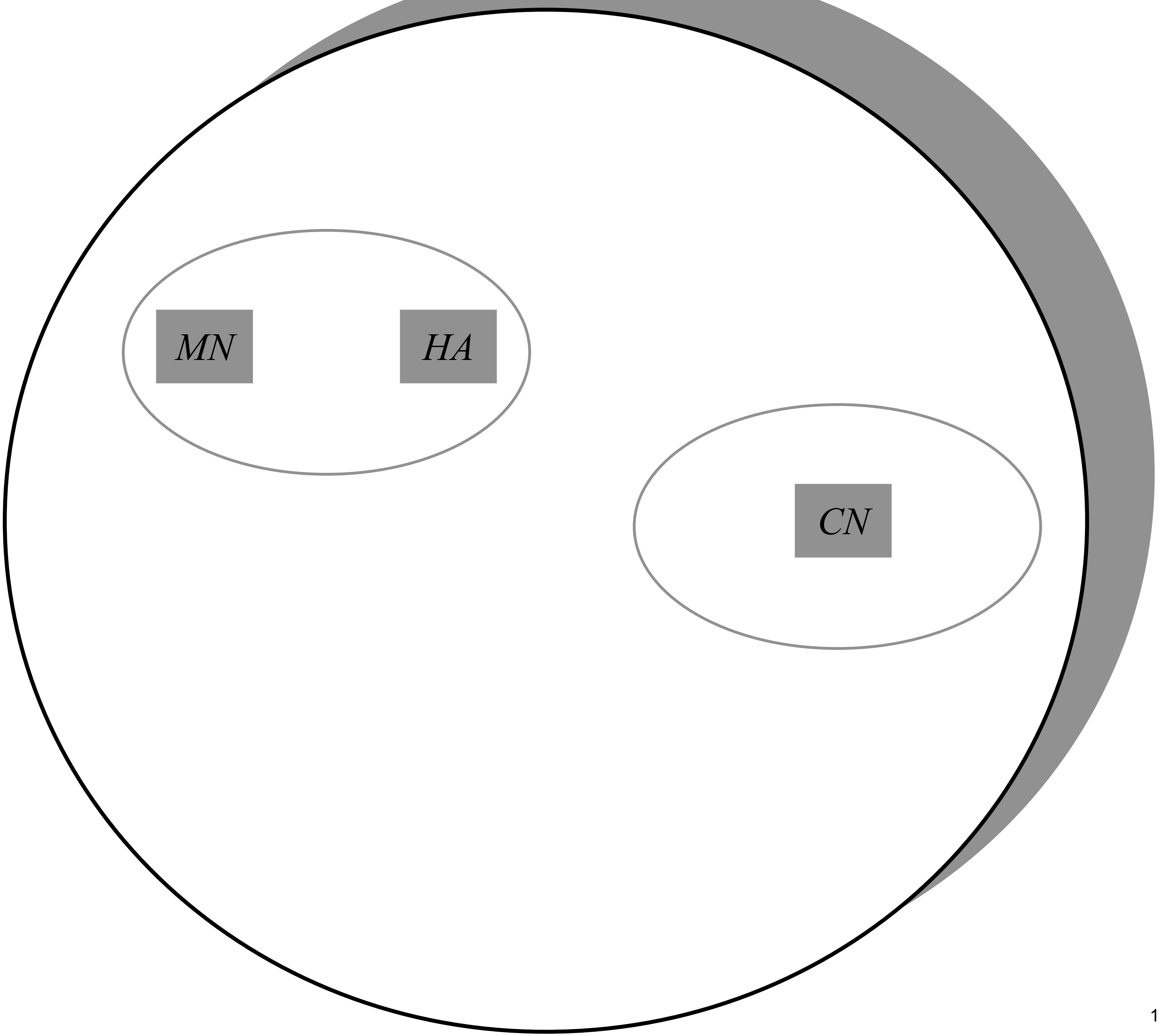
Mobile IPv6

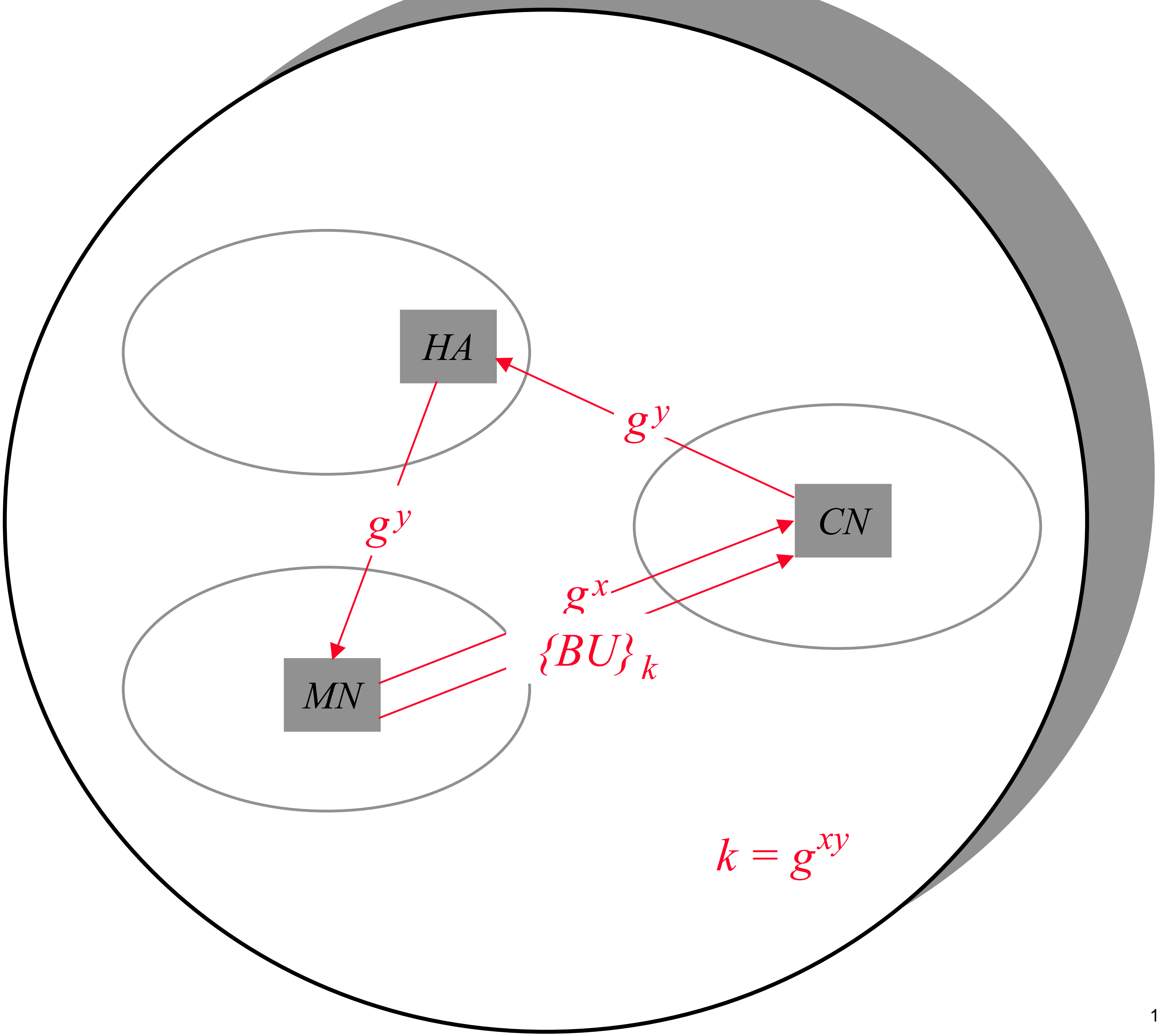
- home address
  - the node is always addressed by the same IP number
- care-of addresses (one or more)
  - bind dynamically to different subnet IP numbers
    - » all packets containing the binding information must be authenticated
    - » authentication relies upon previously established security associations
- Binding Update/Acknowledgement
  - realized through Destination Options Headers
  - Binding Cache integrated with Destination Cache

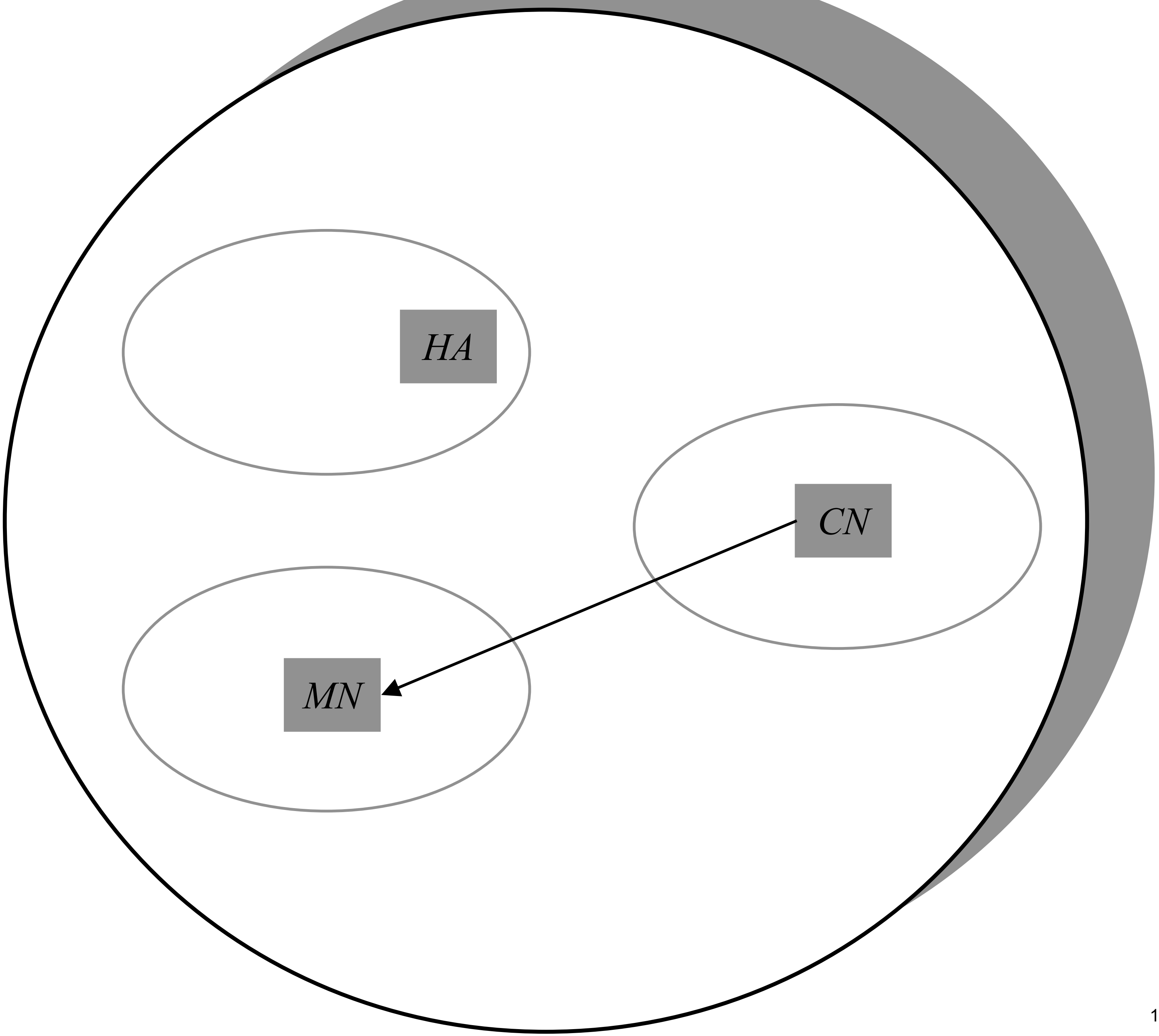
Mobile IPv6 proposal

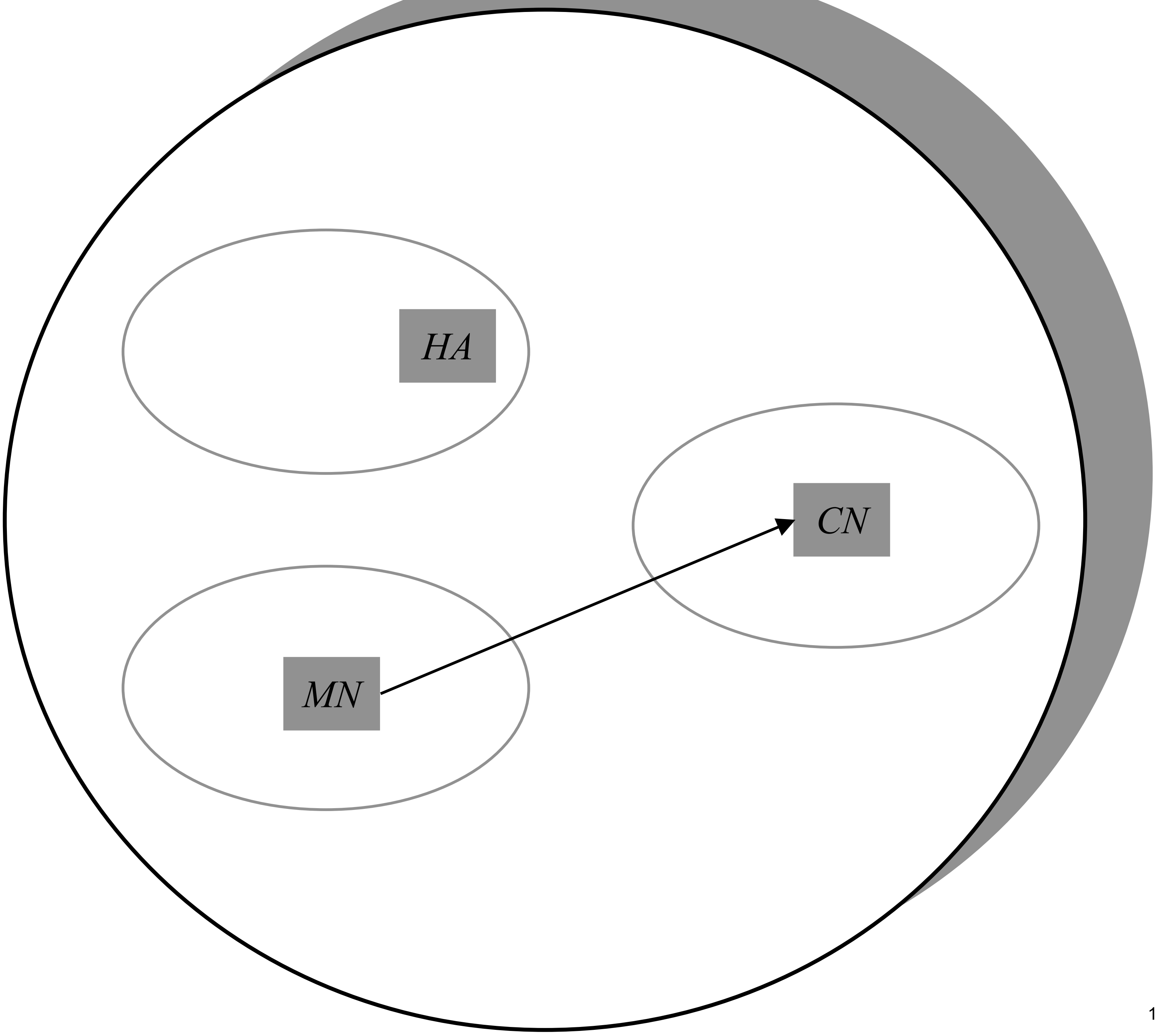


initial architecture

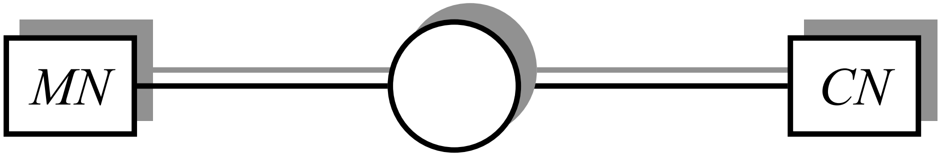

MN
HA
CN


HA
g^y
g^y
CN
g^x
MN
$\{BU\}_k$
$k = g^{xy}$


HA
CN
MN

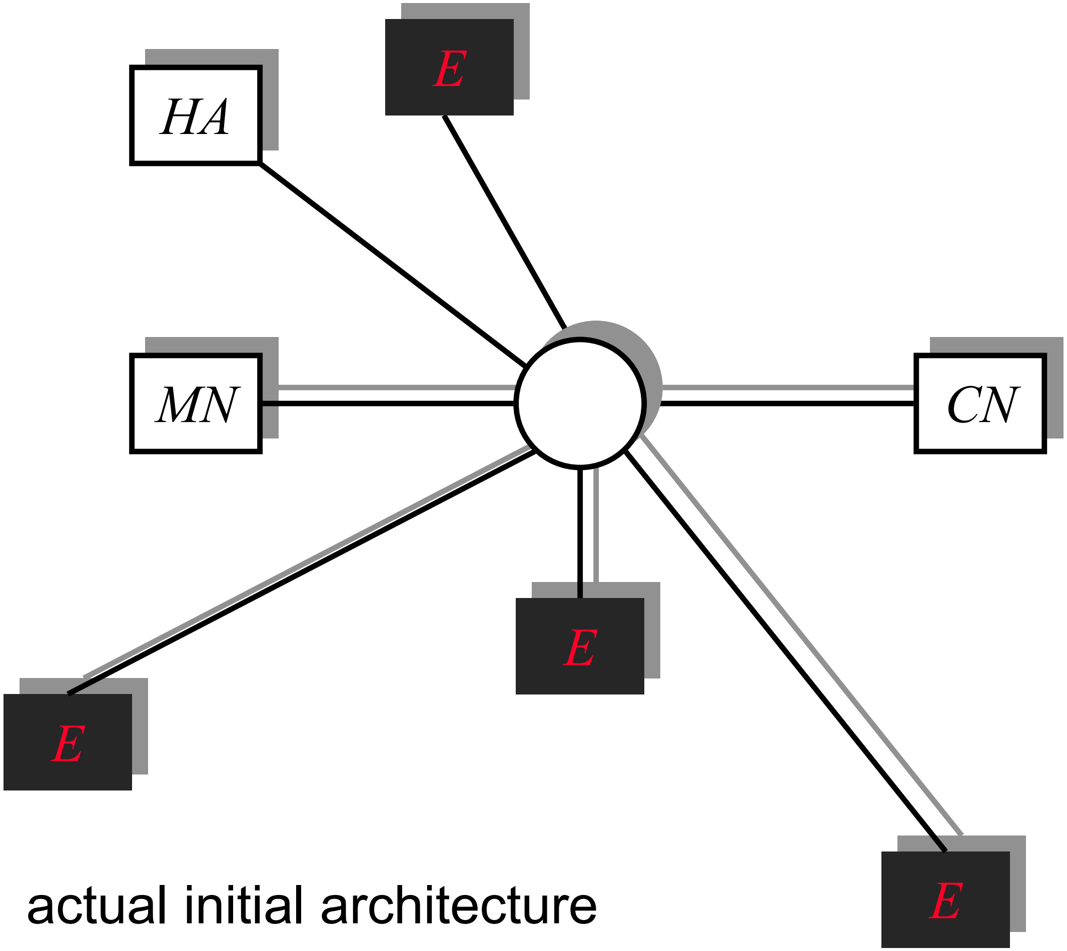

HA
CN
MN

Mobile IPv6 proposal

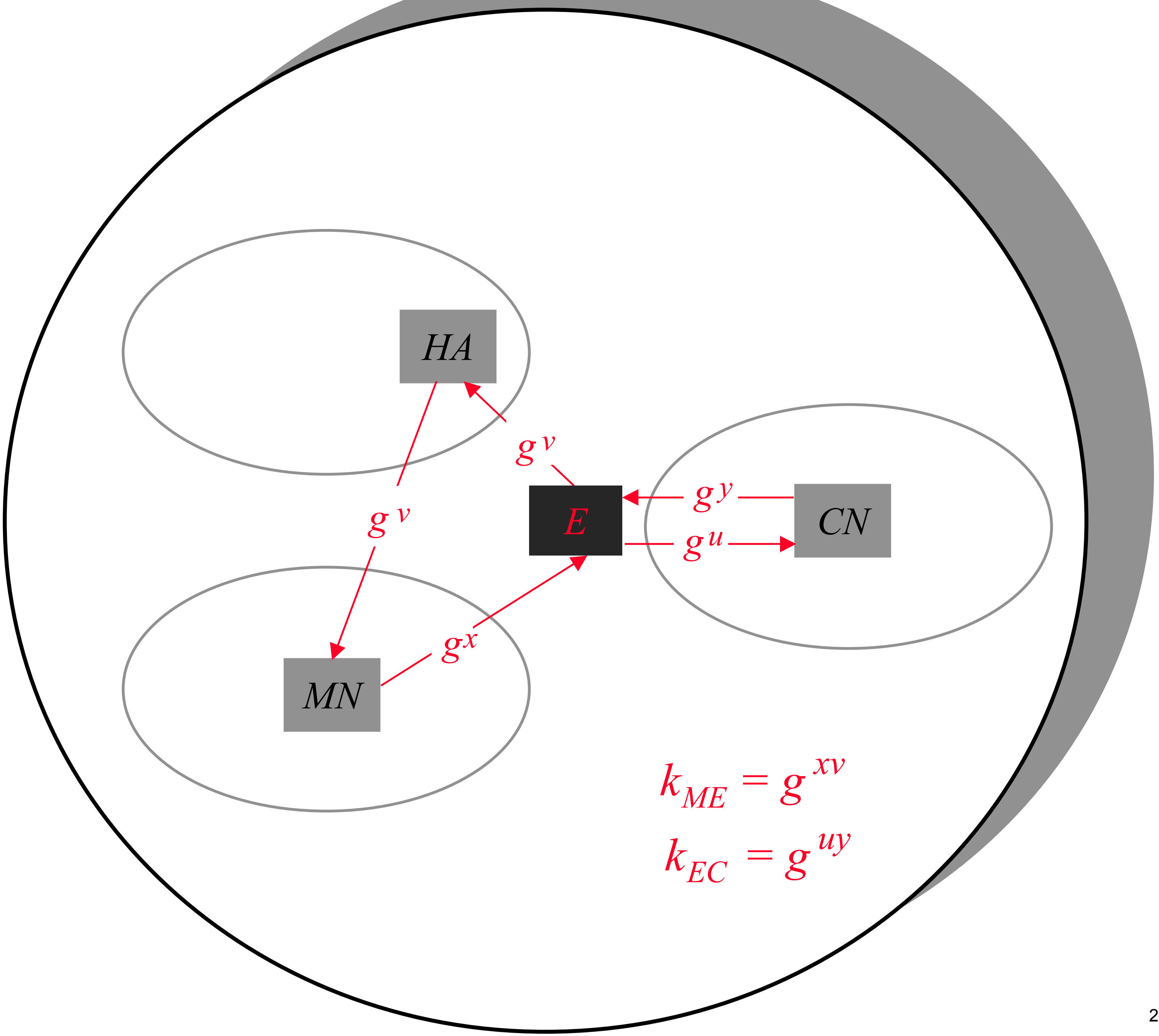


session architecture

Mobile IPv6 proposal



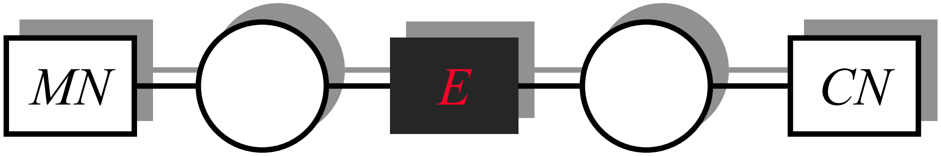
actual initial architecture



$$k_{ME} = g^{xv}$$

$$k_{EC} = g^{uy}$$

Mobile IPv6 proposal



possible session architecture

Task

Use especs

to add authentication!

Task

- Assess tradeoff between
  - maximizing strength of authentication
  - minimizing need for infrastructure

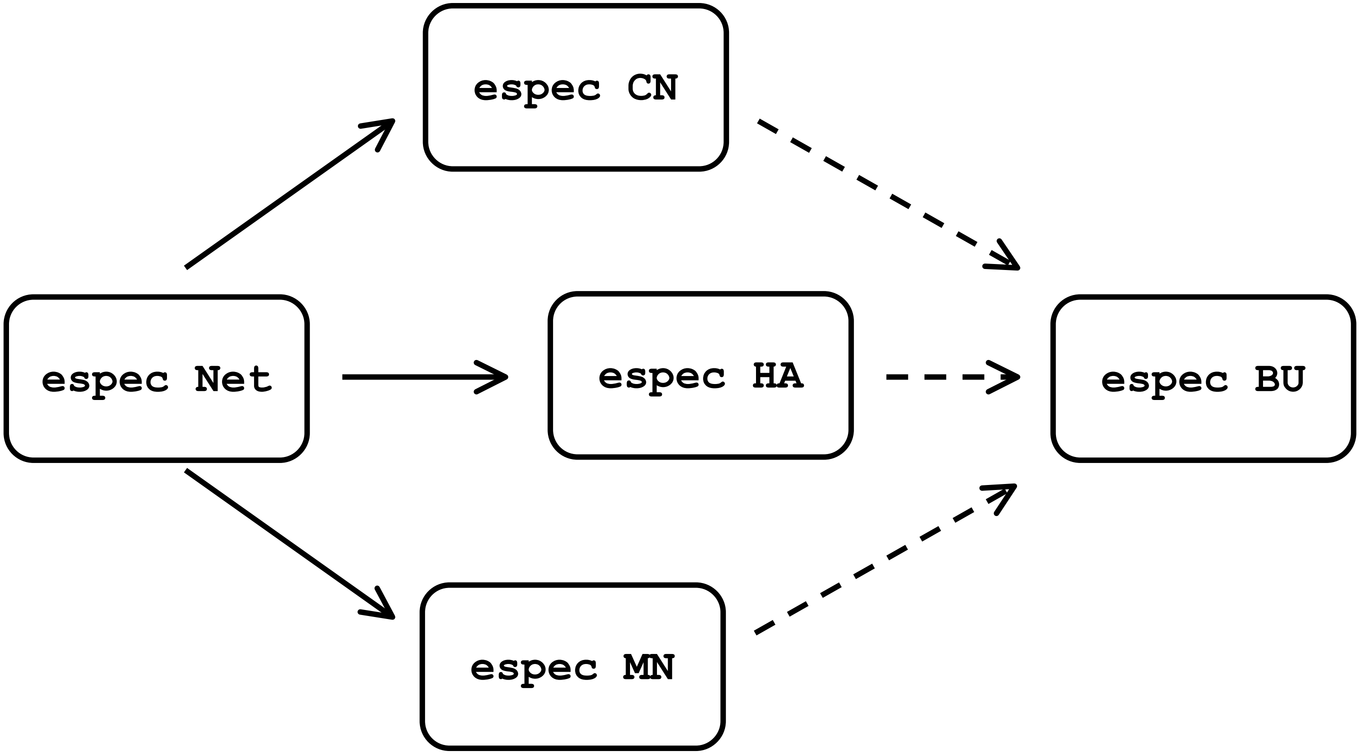
MN’s view

```
espec MN
```

CN’s view

```
espec CN
```

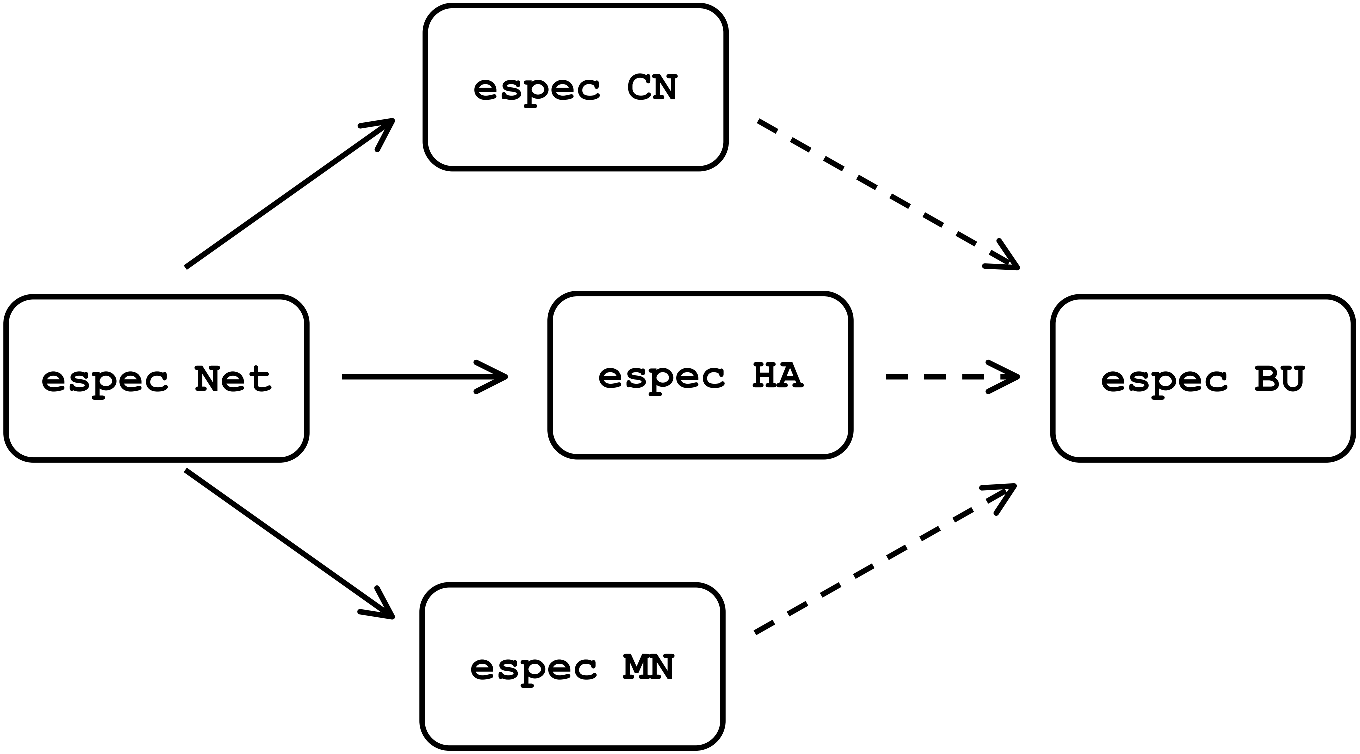
BU architecture



(aspects of especs)

- genericity
  - all agents are instances of cord espec
- automated
  - composition of agents
  - trace generation
- support for formal analysis
  - model checking
  - theorem proving
  - invariant generation

BU architecture



BU architecture

diag BU

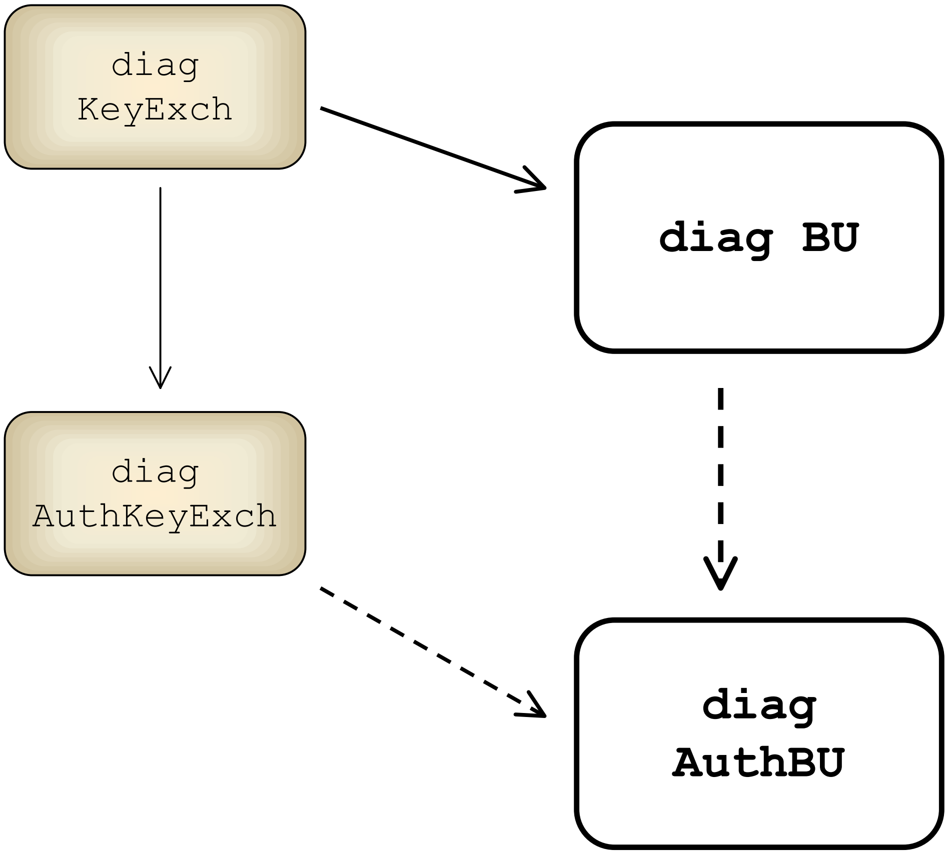
(aspects of especs)

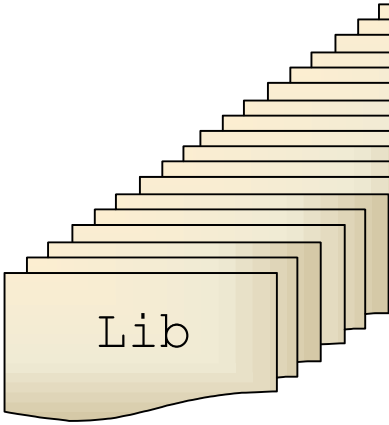
- adjustable abstraction level
- stratification:
  - agents: process calculus
  - protocols: especs
  - architectures: diagrams
    - network connectors and components
    - infrastructure and chain of trust
    - information flow
    - …

BU architecture

`diag BU`

BU refinement

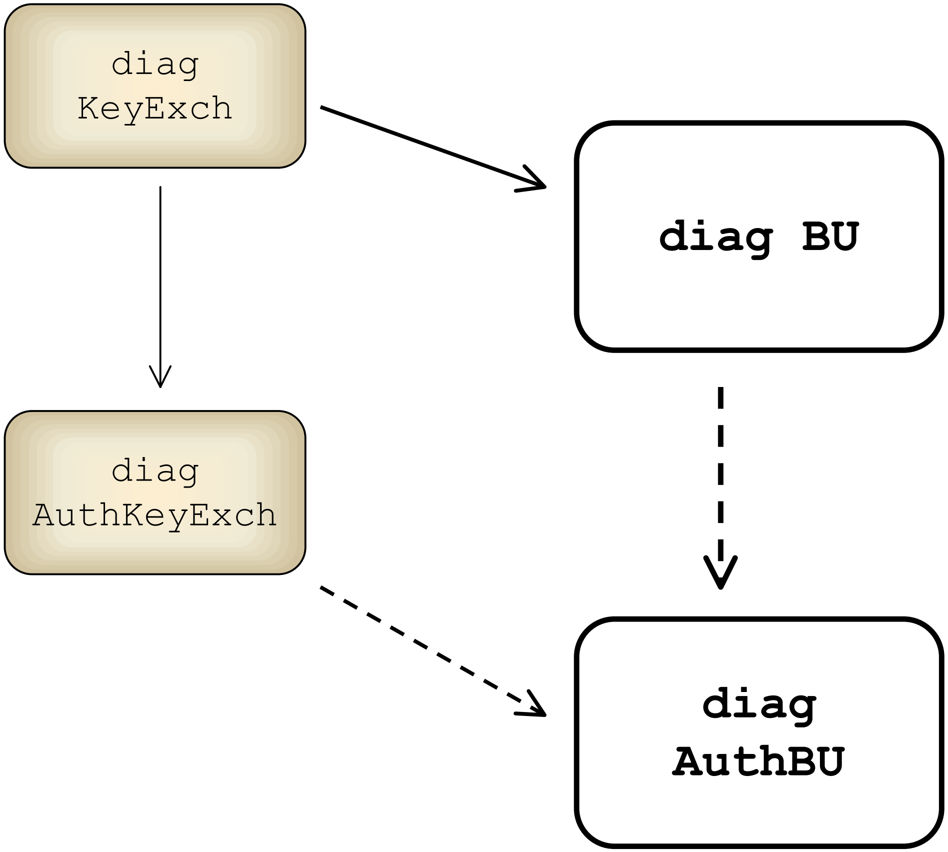


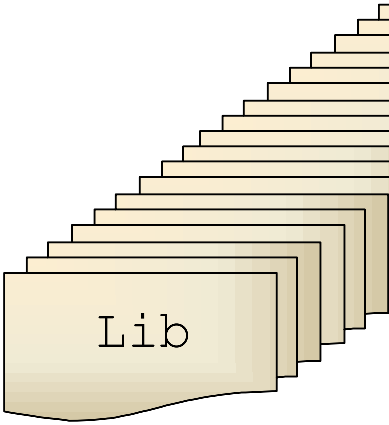


(aspects of especs)

- development (programming, generation)
  - – top-down: refinement
    - » morphisms: inheritance, genericity
  - – bottom-up: composition
    - » pushouts
    - » emergent and vanishing properties
    - » game theory, linear logic (strategies)
  - – program transformation
    - » authentication compiler (Bellare-Canetti-Krawczyk)
    - » optimization
  - – adaptation
    - » specification-carrying software

BU refinement





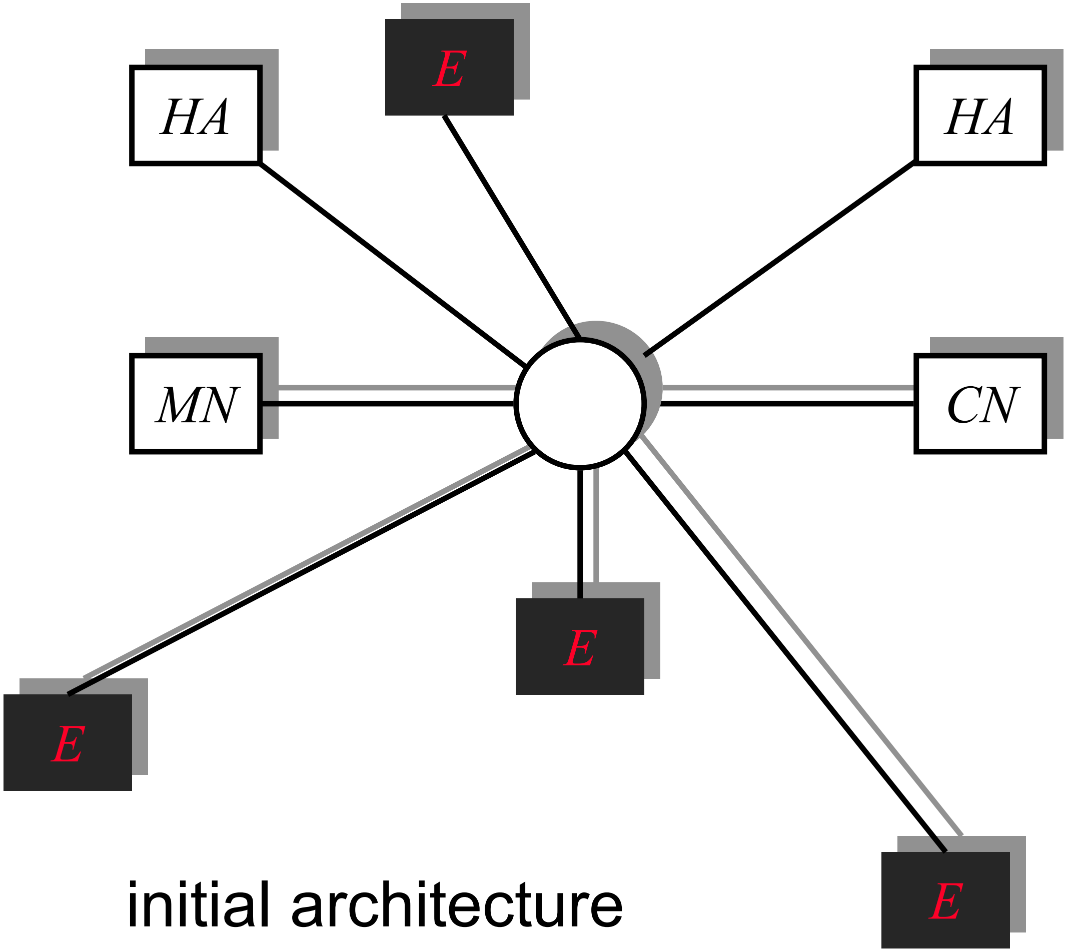
AuthBU architecture

diag
AuthBU

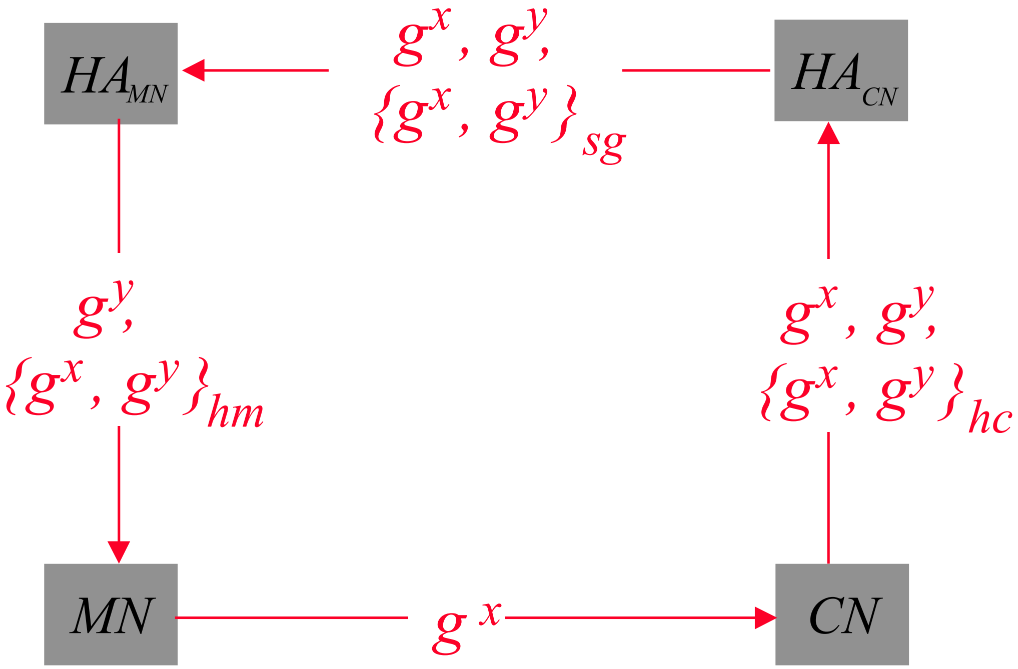
AuthMN’s view

```
espec
AuthMN
```

Authenticated MIPv6



initial architecture



HA_{MN}

HA_{CN}

MN

CN

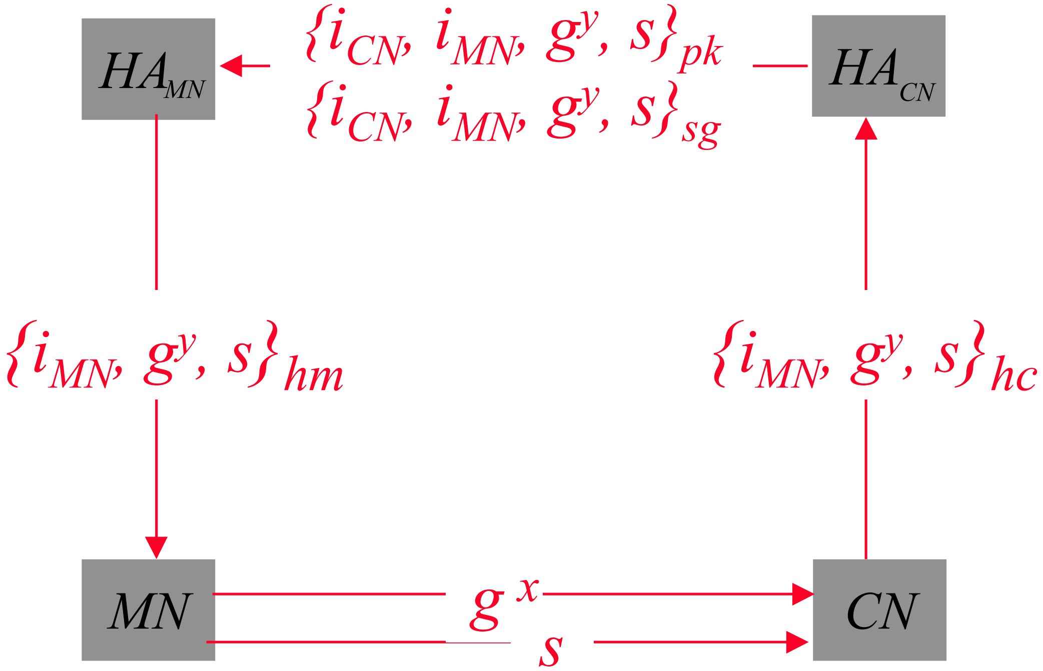
g^x

$g^x, g^y, \{g^x, g^y\}_{hc}$

$g^x, g^y, \{g^x, g^y\}_{sg}$

$g^y, \{g^x, g^y\}_{hm}$

$$k = g^{xy}$$



HA_{MN}

HA_{CN}

$\{i_{CN}, i_{MN}, g^y, s\}_{pk}$

$\{i_{CN}, i_{MN}, g^y, s\}_{sg}$

$\{i_{MN}, g^y, s\}_{hm}$

$\{i_{MN}, g^y, s\}_{hc}$

MN

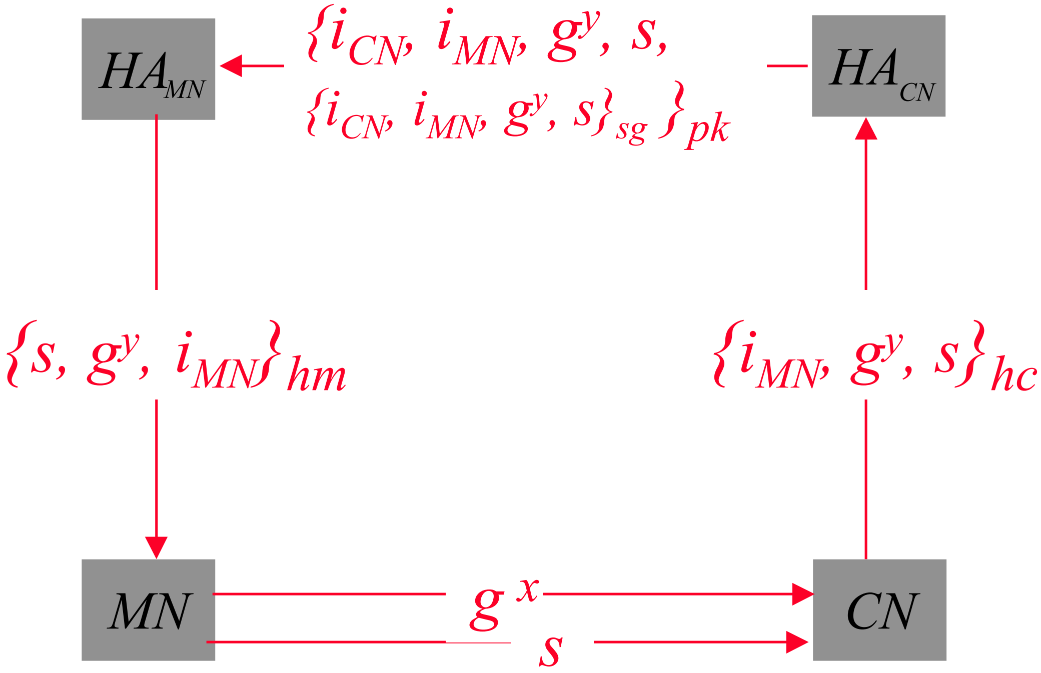
CN

g^x

s

$k = g^{xy}$

$s = \{i_{CN}, i_{MN}, g^x, g^y\}_k$



HA_{MN}

HA_{CN}

$\{i_{CN}, i_{MN}, g^y, s, \{i_{CN}, i_{MN}, g^y, s\}_{sg} \}_{pk}$

$\{s, g^y, i_{MN}\}_{hm}$

$\{i_{MN}, g^y, s\}_{hc}$

MN

CN

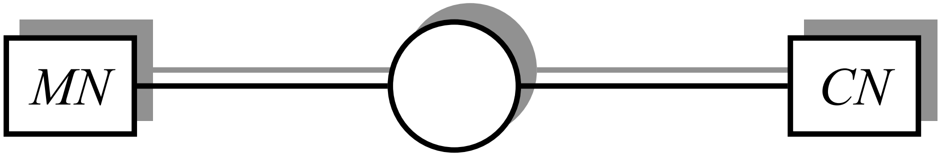
g^x

s

$$k = g^{xy}$$

$$s = \{i_{CN}, i_{MN}, g^x, g^y\}_k$$

Authenticated MIPv6



assured session architecture

Variations

- weaker authentications:
  - one-way: no PKI, just certificates, or AAA - no anonymity
  - first time unauthenticated (like SSH), then chained hashing
- stronger authentications:
  - privacy
  - anonymity, non-repudiation
- dynamic infrastructure
  - no shared secret: databases of “fingerprints”
  - authenticating by non-forgeable capability
  - authenticating by divided secret

(aspects of especs)

- additional aspects:
  - – information flow
  - – information hiding
  - – cryptography
  - – …

Ongoing work

IMPLEMENT the tool!

Papers

- Authentication for Mobile IPv6
  - with A. Datta, J. Mitchell and F. Muller
- Composition and refinement of behavioral specifications
  - with D. Smith
- Guarded transitions in evolving specifications
  - with D. Smith

http://www.kestrel.edu/users/pavlovic/

(cord spaces)

(names) $N ::= X \mid A$

(terms) $t ::= x \mid a \mid N \mid t,\ldots,t \mid {}_N\{t\}$

(actions) $a ::= \langle t\rangle \mid (x) \mid (t/p(x))$

(strands) $S ::= aS$

(cords) $C ::= [S]_{\approx}$

$FV(t) = \emptyset$

(interaction) $[(x)R] \otimes [\langle t\rangle S] \ldots \quad \rhd\rhd \quad [R(t/x)] \otimes [S] \ldots$

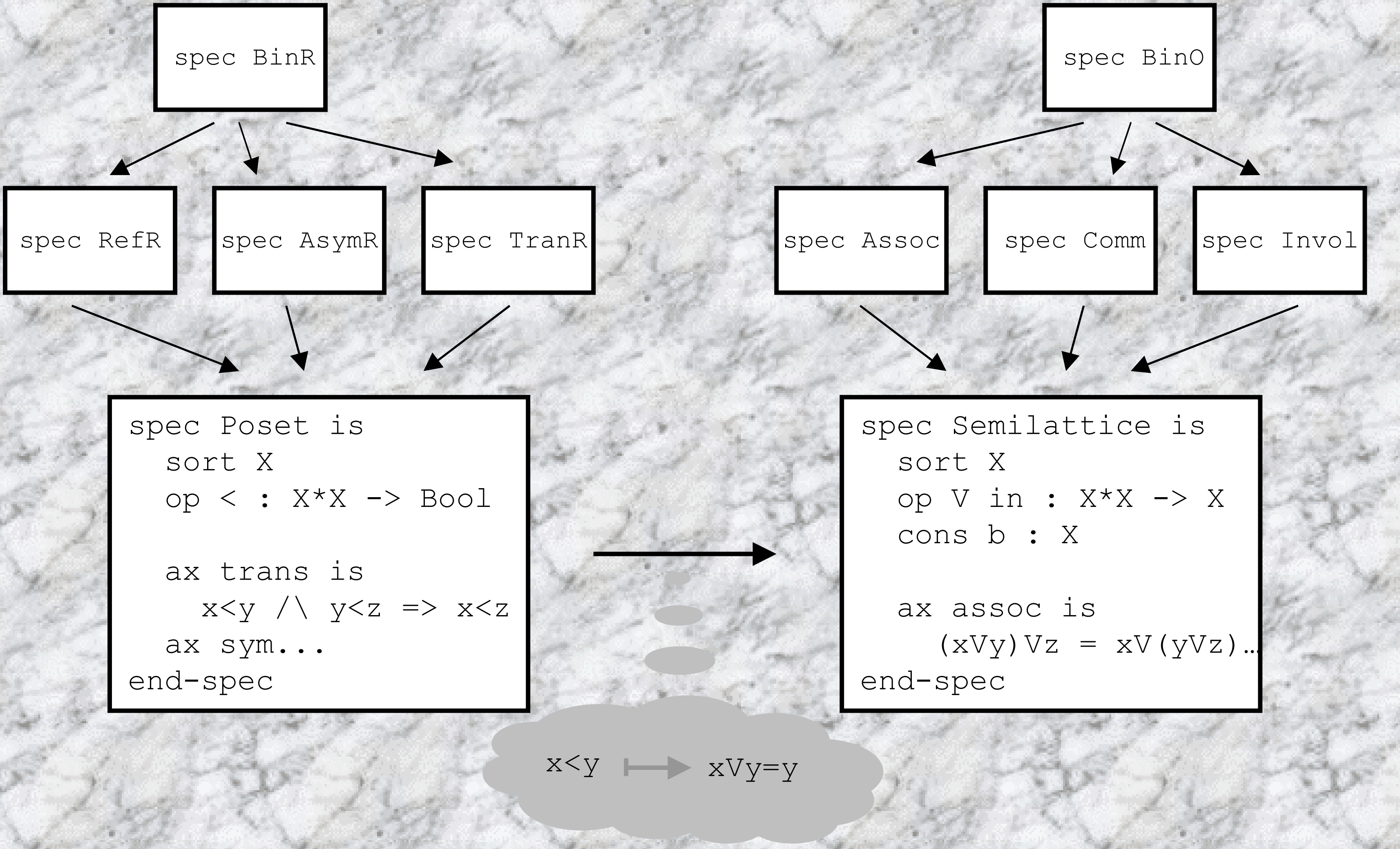
(reaction) $[(p(t)/p(x))R] \ldots \quad \rhd\rhd \quad [R(t/x)] \ldots$

$FV(t) = \emptyset$

What are especs?

- diagrams of specs
- specification-carrying programs
- in a development environment supporting
  - refinement (top-down)
  - composition (bottom-up)
  - synthesis of verified code
- programming language with
  - guarded commands
  - logical annotations as first-class citizens (available at runtime)
  - procedural abstraction and refinement

What are specs?



spec BinR

spec RefR

spec AsymR

spec TranR

```
spec Poset is
  sort X
  op < : X*X -> Bool

  ax trans is
    x<y /\ y<z => x<z
  ax sym...
end-spec
```

spec BinO

spec Assoc

spec Comm

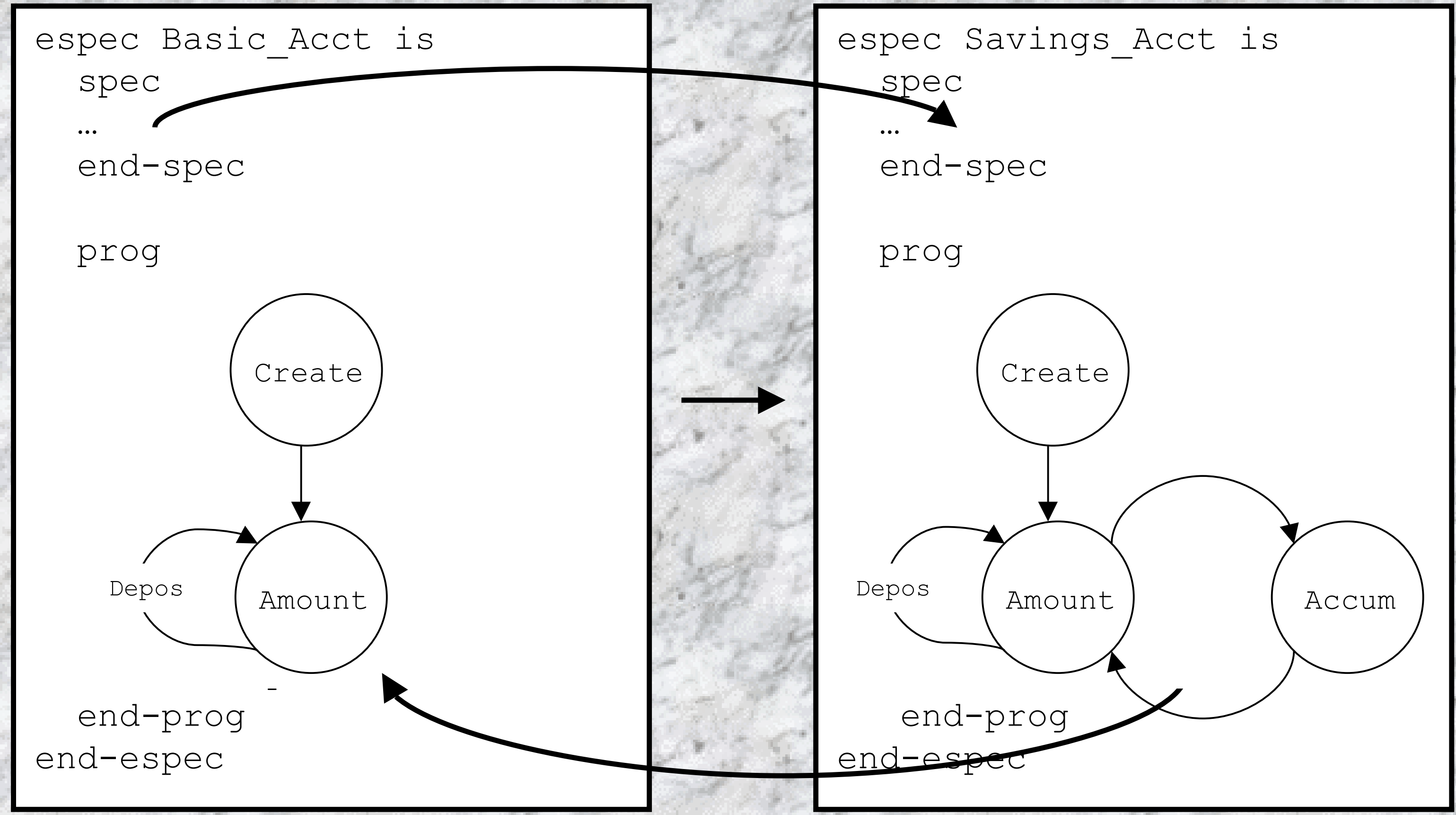
spec Invol

```
spec Semilattice is
  sort X
  op V in : X*X -> X
  cons b : X

  ax assoc is
     (xVy)Vz = xV(yVz)…
end-spec
```

x<y ⟼ xVy=y

What are especs?



```
espec Basic_Acct is
  spec
  …
  end-spec

  prog

  end-prog
end-espec
```

```
espec Savings_Acct is
  spec
  …
  end-spec

  prog

  end-prog
end-espec
```