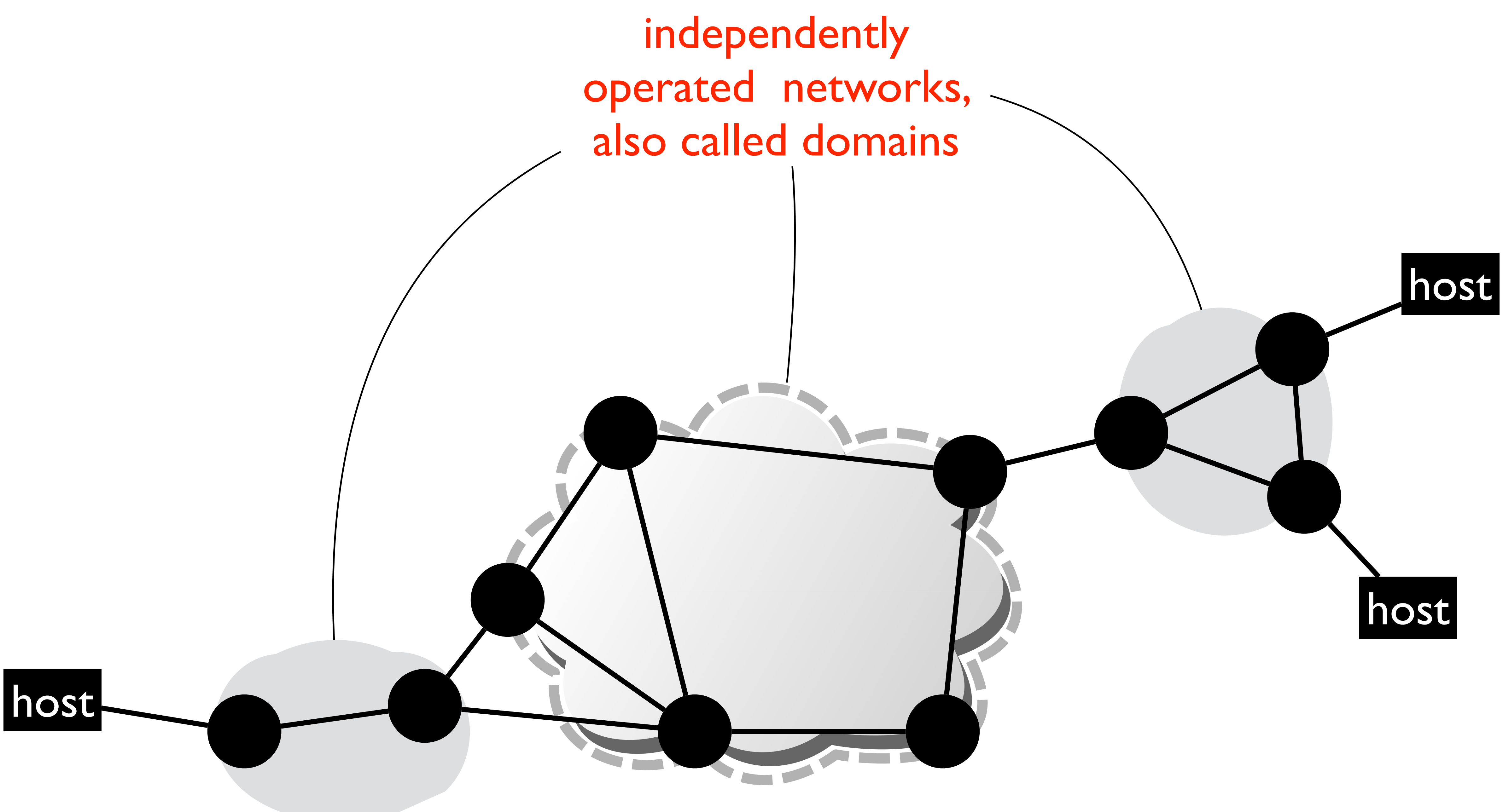
A Logical Approach to Representing and Reasoning About Interdomain Routing Policies

Anduo Wang and Zhijia Chen
Temple University

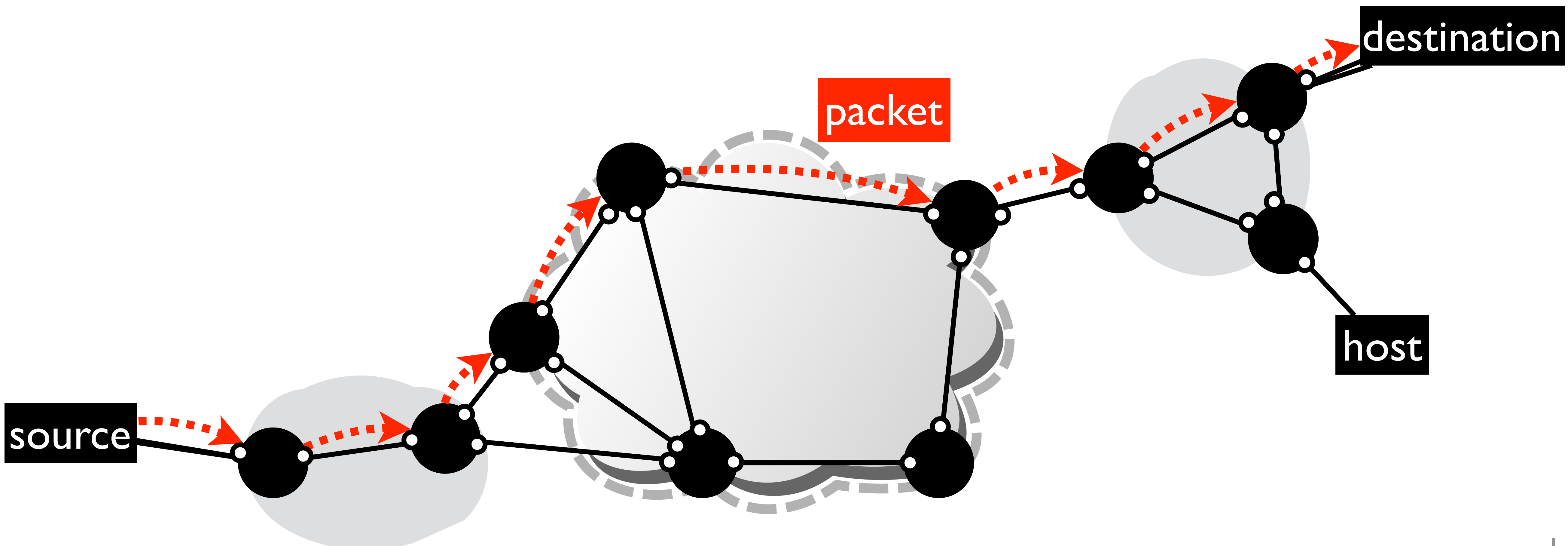
the Internet

a loose federation of networks, each acting in their own self interests — utilization, performance …



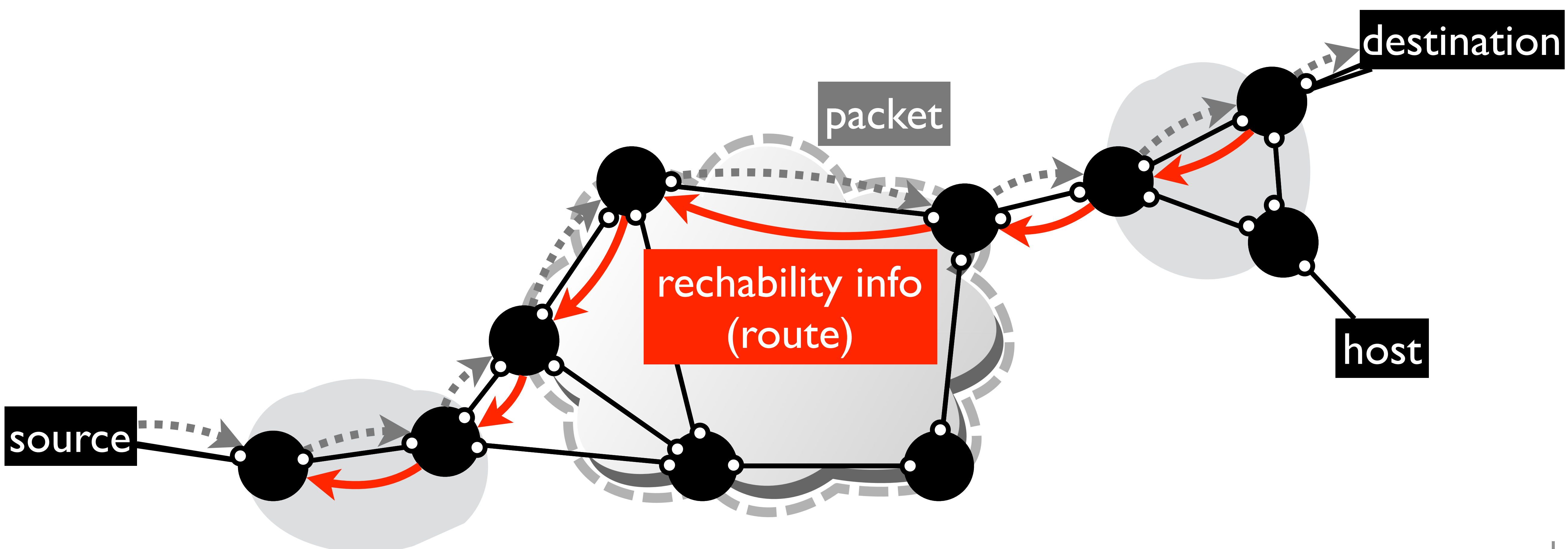
interdomain routing

determine a sequence of domains and routers a packet will traverse in passing from the source to the destination



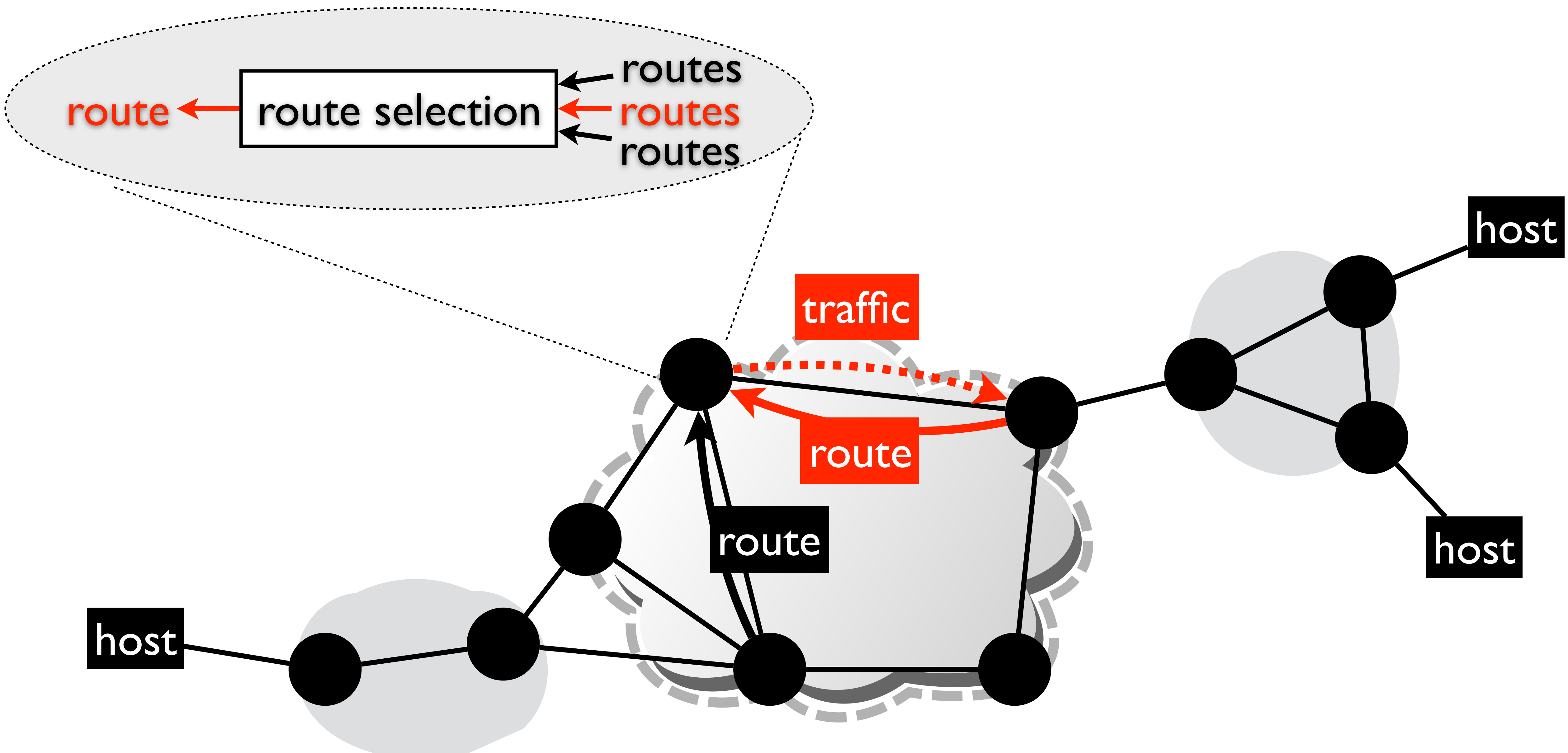
interdomain routing

determine a sequence of domains and routers a packet will traverse in passing from the source to the destination



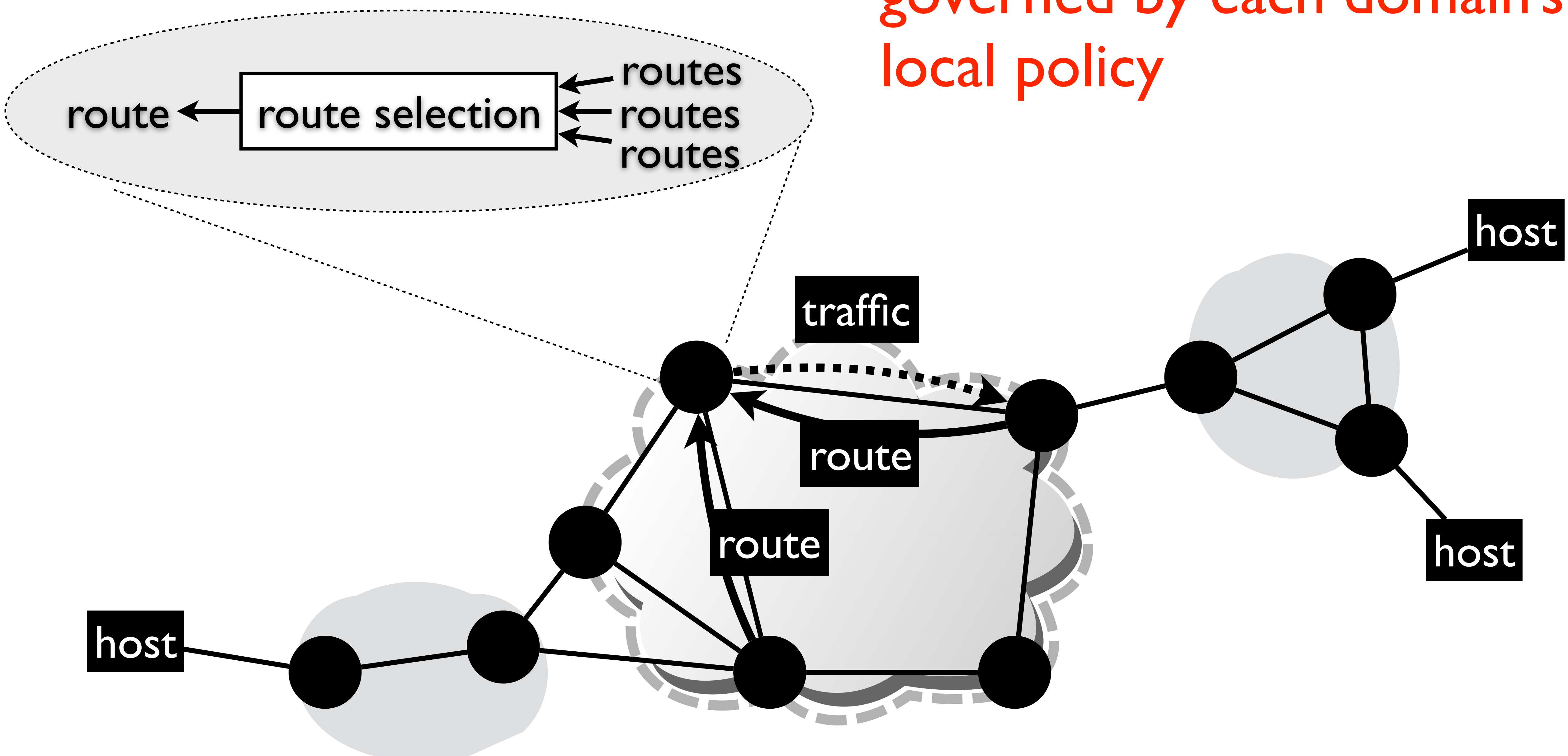
interdomain routing

one best route is selected out of all available routes based on some measure of the route



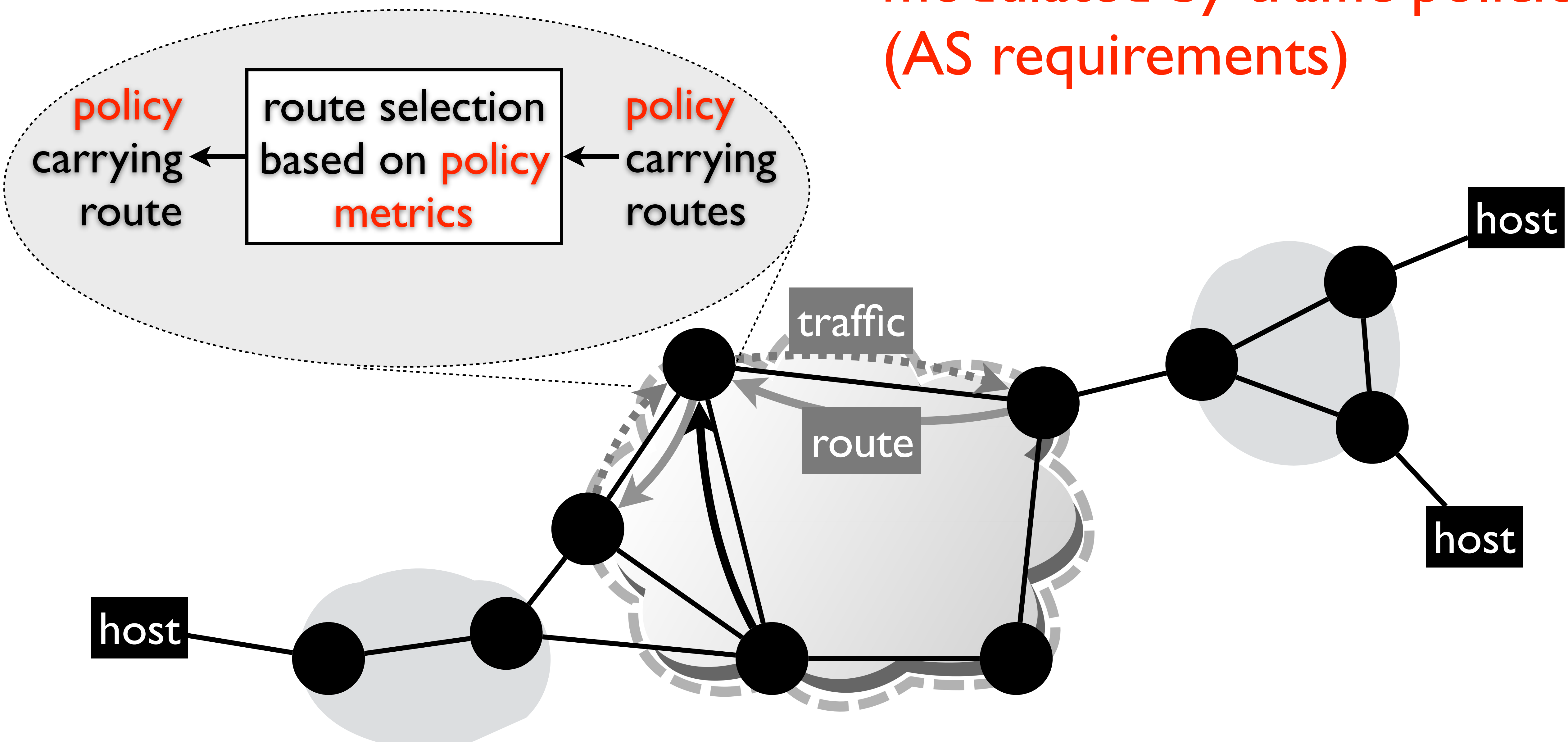
interdomain routing

one best route is selected out of all available routes based on some measure of the route — governed by each domain's local policy



policy routing

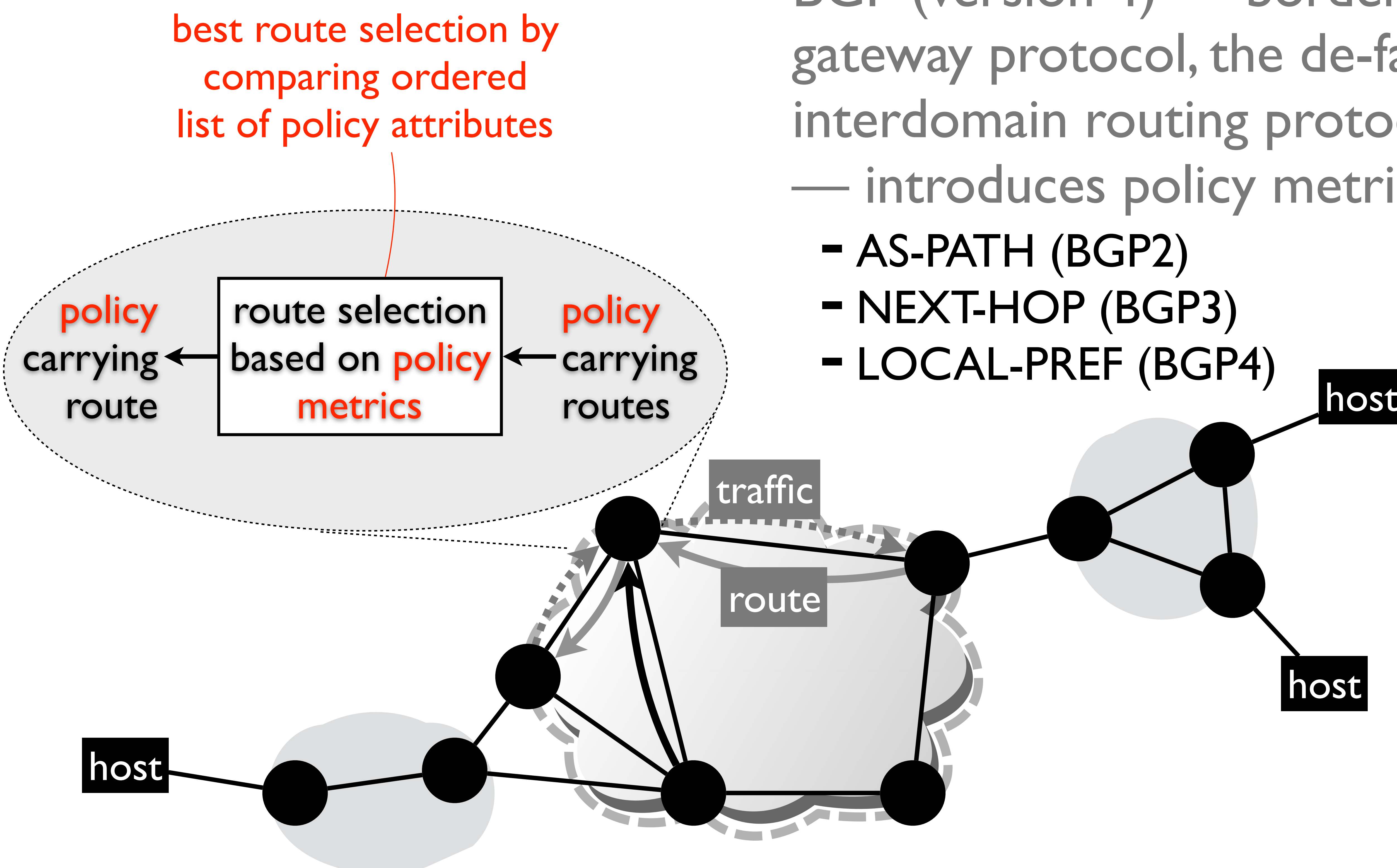
maintains inter-domain connectivity — "shortest path" to any destination — as modulated by traffic policies (AS requirements)



today's Internet — policy driven

BGP (version 4) — border gateway protocol, the de-facto interdomain routing protocol — introduces policy metrics

- AS-PATH (BGP2)
- NEXT-HOP (BGP3)
- LOCAL-PREF (BGP4)



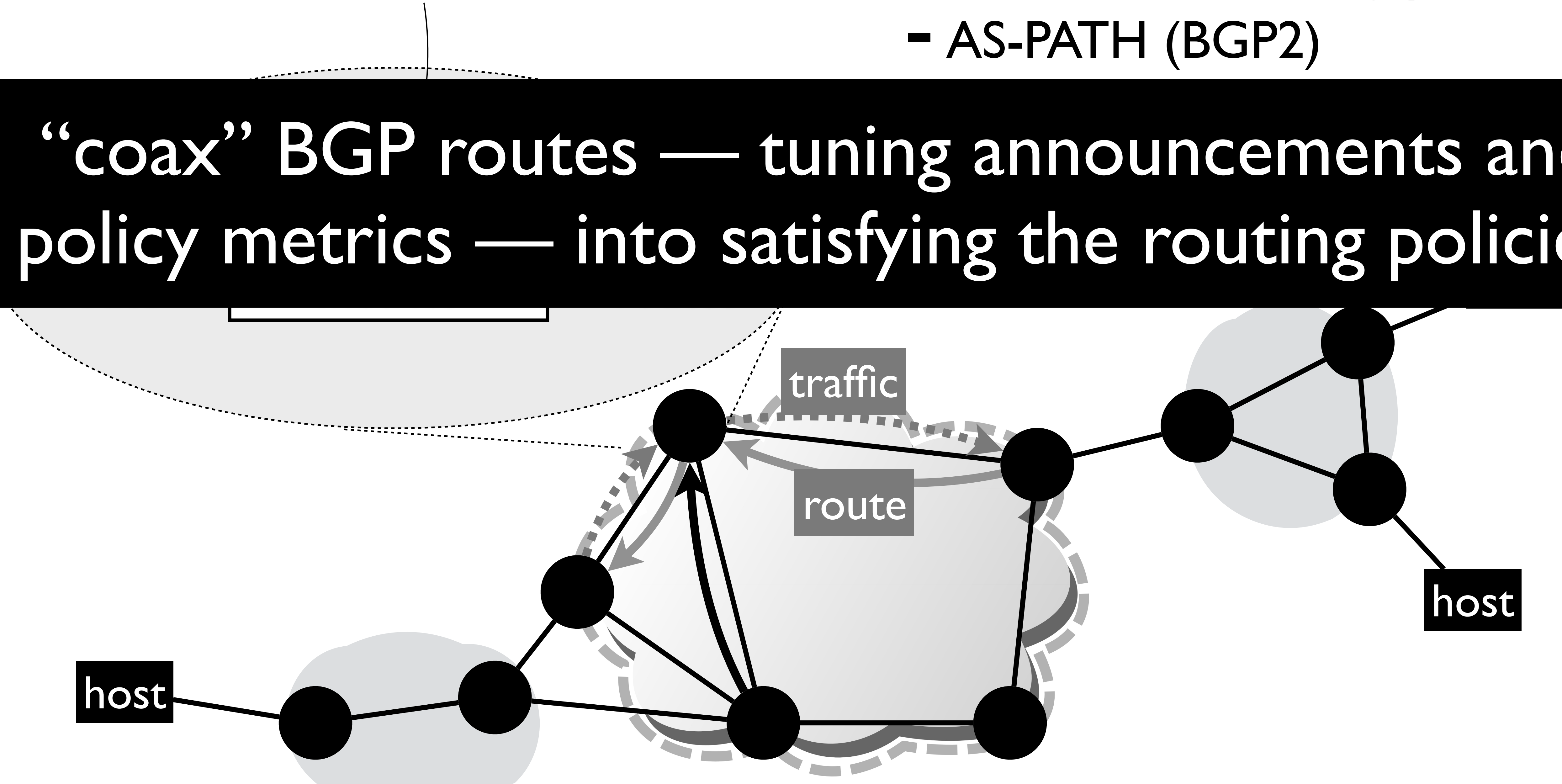
today’s routing policies — *buried in BGP*

best route selection by
comparing ordered
list of policy attributes

BGP — border gateway protocol, the de-facto interdomain routing protocol

- AS-PATH (BGP2)

“coax” BGP routes — tuning announcements and policy metrics — into satisfying the routing policies

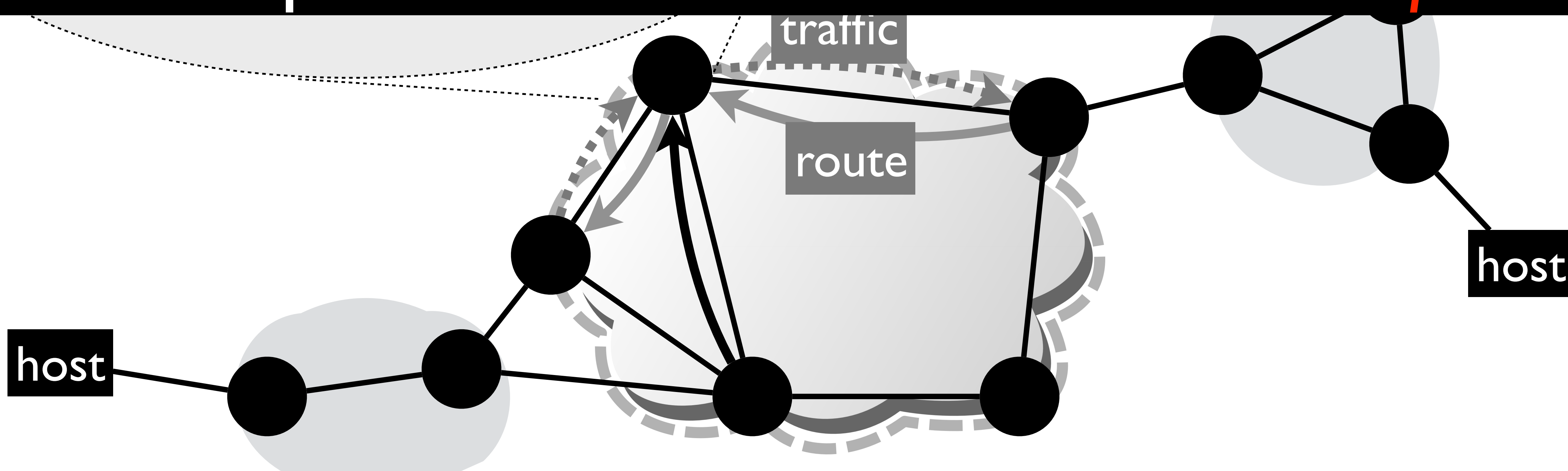


today's routing policies — *operational*

best route selection by comparing ordered list of policy attributes

BGP — border gateway protocol, the de-facto interdomain routing protocol

- representation — policies buried in policy metrics are *indirect and low level*
- coordination — policies refactored into prefixed ordered path metrics are *manual and error-prone*



this talk

can we take a more principled logical approach towards routing policies, making policies easier to understand and combine

a declarative approach

a unifying representation

- policies as data integrity constraints

enabling automated coordination

- reasoning about policy interactions

a unifying representation

explicit data abstraction, unifying a wide range of policies previously buried in the routing protocol

- network state as relations and rules
- network policies as data integrity constraints (ICs)

network state

factual network state as relations and rules `N=IUR`

- `I`, a finite set of ground facts

example

- incoming route relation $r_i/3$, outgoing relation $r_o/3$,
- three attributes `destination`, `next_hop`, and `path_vector`

```
%% Network ground facts
%% 3 incoming routes and 2 outgoing routes
I1: ri(1.2.3.4, 'router1', [AS2,AS3,AS5]) :-
I2: ri(1.2.3.4, 'router2', [AS2,AS4,AS5]) :-
I3: ri(1.2.3.5, 'router1', [AS3,AS5]) :-
I4: ro(1.2.3.4, 'router2', [AS2,AS4,AS5]) :-
I5: ro(1.2.3.5, 'router1', [AS4,AS5]) :-
```

network state

a factual network state `N=IUR`

- `R`, derived network knowledge by rules

example

- all paths to a particular destination (`1.2.3.4`)

```
%% Derived path information
%% all available paths to a 1.2.3.4
R1: rpath(z) :- ri(x,y,z), x=1.2.3.4.
```

network policy

policies as a finite set of integrity constraints (ICs)

- *generative* form and *denial* form

example — path validity

- any selected outgoing route must correspond to some incoming route

```
%% path validity policy
%% generative form
IC_validity: r_i(x,y,z) :- r_o(x,y,z).

%% denial form
IC_validity': :- r_o(x,y,z), ¬r_i(x,y,z).
```

policy expressiveness

non-aggregate policy

- constraints over a single path

example — explicit path (ep)

- to better control end to end performance, a sender host may want to specify the <u>explicit path (a)</u> for carrying traffic to a certain destination (d)

```
%% explicit path policy
IC_ep: z=a :- r_o(x,y,z), x=d.
```

policy expressiveness

non-aggregate policy

- constraints over a single path

example — business relationship guideline (GR)

- to maximize revenue and minimize cost, regulate route selection based on business relations — prefer a route from a customer (respectively, peer) route over a provider route

```
%% Gao-Rexford Policy Guideline
IC_GR1 :-r_o(x,y,z),r_i(x,y′,z′),provider(z),customer(z′).
IC_GR2 :-r_o(x,y,z),r_i(x,y′,z′),provider(z),peer(z′).
```

policy expressiveness

non-aggregate policy

- constraints over a single path

example — (MIRO) avoiding unsafe ASes

- MIRO is an extension to today's interdomain routing, allowing networks to negotiate paths

```
%% negotiates a route bypassing a suspicious node b
IC_MIRO :- r_o(x,y,z), waypoint(z,b).
```

policy expressiveness

non-aggregate policy

- constraints over a single path

aggregate policy

- involve a group of routes
- *without* explicit use of aggregate term

policy expressiveness

aggregate policy

- involve a group of routes
- *without* explicit use of aggregate term

example — shortest path (sp)

- select route that has the fewest (AS) hops

```
%% shortest path
IC_sp :- r_o(x,y,z), r_i(x,y2,z2), length(z)>length(z2).
```

policy expressiveness

aggregate policy

- involve a group of routes
- *without* explicit use of aggregate term

example — (WISER) joint traffic engineering

- WISER is an extension to the Internet that allows neighboring ASes to jointly select a path that has the lowest overall cost

```
%% WISER policy
R_Wiser  j(x,y,z) :- Local(x,y,z1),Advertised(y,z2),z=z1+z2.
IC_Wiser :- ro(x,y,z), j(x,y,w), j(x,y2,w2), w>w2.
```

policy expressiveness

ICs unify popular policies and futuristic ones

- non-aggregate policy
  - constraints over a single path
- aggregate policy
  - involve a group of routes
  - *without* explicit use of aggregate term

automated coordination

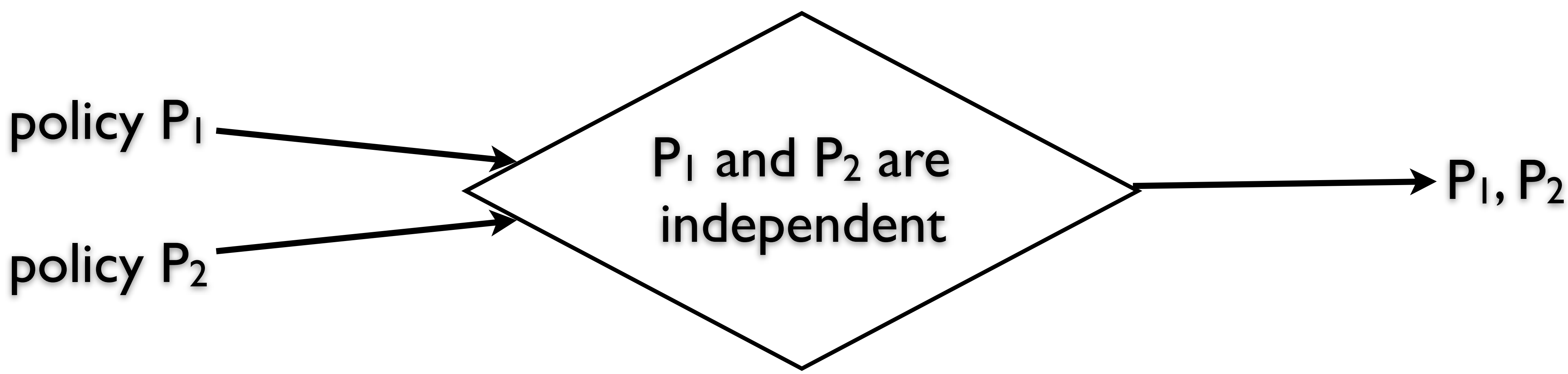
advantage of logic representation — coordination by automated reasoning

- determine the interactions between the policies
- combine policies into a coherent new whole

coordination

advantage of logic representation — coordination by automated reasoning

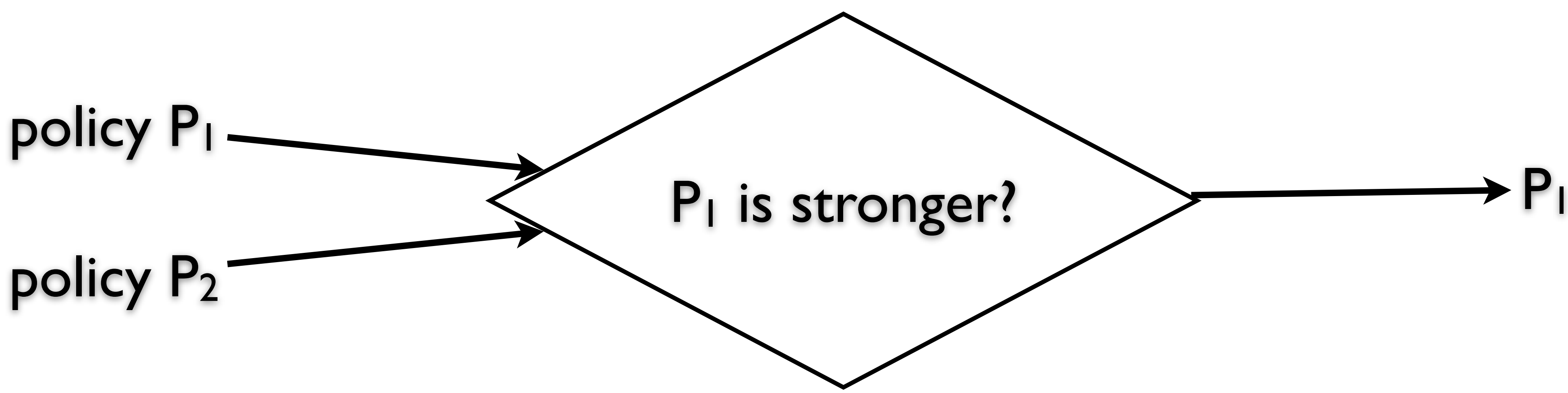
- determine the interactions between the policies
- combine policies into a coherent new whole



coordination

advantage of logic representation — coordination by automated reasoning

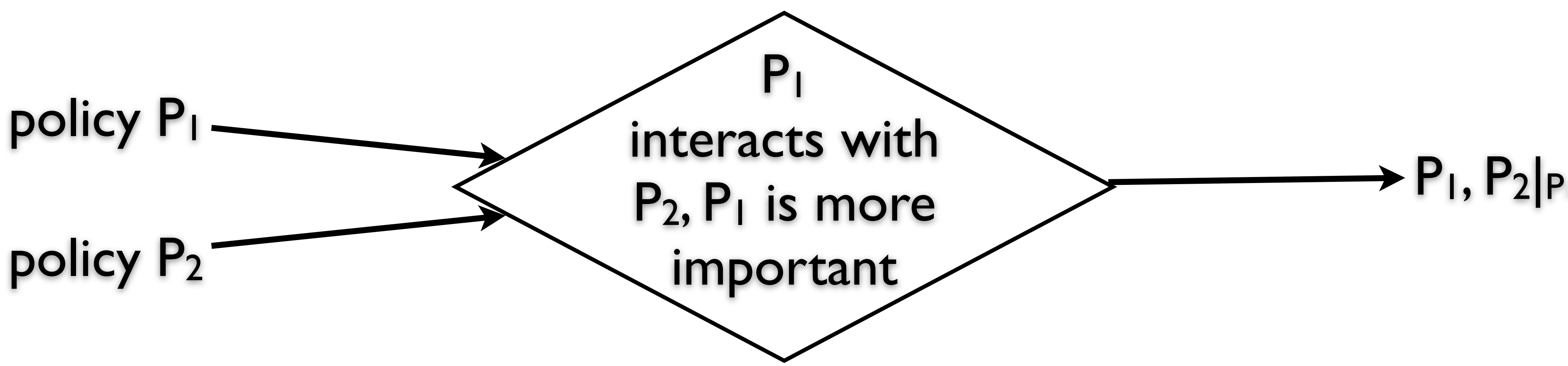
- determine the interactions between the policies
- combine policies into a coherent new whole



coordination

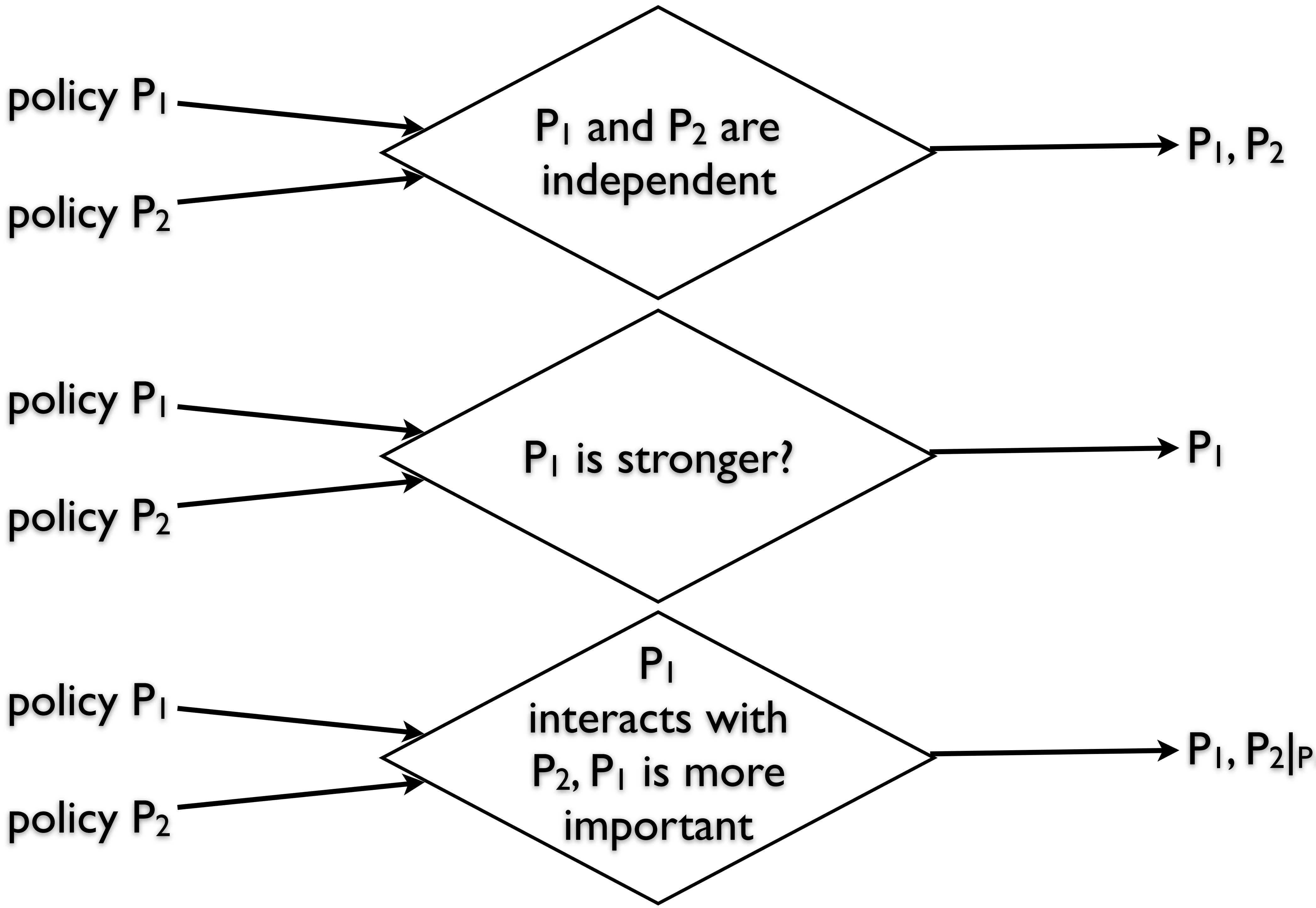
advantage of logic representation — coordination by automated reasoning

- determine the interactions between the policies
- combine policies into a coherent new whole



coordination

key — derive the impact of P_1 on P_2



a realization with the residue method

residue anticipates the impact of P_1 on P_2

- a fragment of P_1 that interacts with P_2
- obtained by (*partial*) *subsumption*

subsumption

(classic) subsumption

for two clauses P_1, P_2: P_1 subsumes P_2 if there exists a substitution σ such that each literal in $P_1\sigma$ occurs in P_2

subsumption

(classic) subsumption

for two clauses P_1, P_2: P_1 subsumes P_2 if there exists a substitution σ such that each literal in $P_1\sigma$ occurs in P_2

subsumption with arithmetics and comparison

for two policies P_1,P_2 of the form A_1:- $B_1 \wedge C_1$ and A_2 :- $B_2 \wedge C_2$

- B_1, B_2 are conjunctions of relational literals
- C_1,C_2 are conjunctions of comparison and arithmetic formulas

P_1 subsumes P_2 if there exists

- a substitution σ such that each literal in (A_1:- B_1)σ occurs in A_2 :- B_2, and
- a solver for arithmetics and comparison that reduces $\neg C_2 \wedge C_1\sigma$ to False

subsumption

(classic) subsumption

for two clauses P_1, P_2: P_1 subsumes P_2 if there exists a substitution σ such that each literal in $P_1\sigma$ occurs in P_2

subsumption with arithmetics and comparison

for two policies P_1,P_2 of the form A_1:- $B_1 \wedge C_1$ and A_2 :- $B_2 \wedge C_2$

- B_1, B_2 are conjunctions of relational literals
- C_1,C_2 are conjunctions of comparison and arithmetic formulas

P_1 subsumes P_2 if there exists

- a substitution σ such that each literal in $(A_1$:- $B_1)\sigma$ occurs in A_2 :- B_2, and
- a solver for arithmetics and comparison that reduces $\neg C_2 \wedge C_1\sigma$ to False

if P_1 subsumes P_2, then P_1 is stronger —
any network compliant with P_1 also satisfies P_2

partial subsumption

P_1 partially subsumes P_2

- if a subclass of P_1 subsumes P_2, signals policy interaction
- a fragment of P_1 — residue — that actually interacts with P_2 can be computed by the subsumption algorithm

partial subsumption

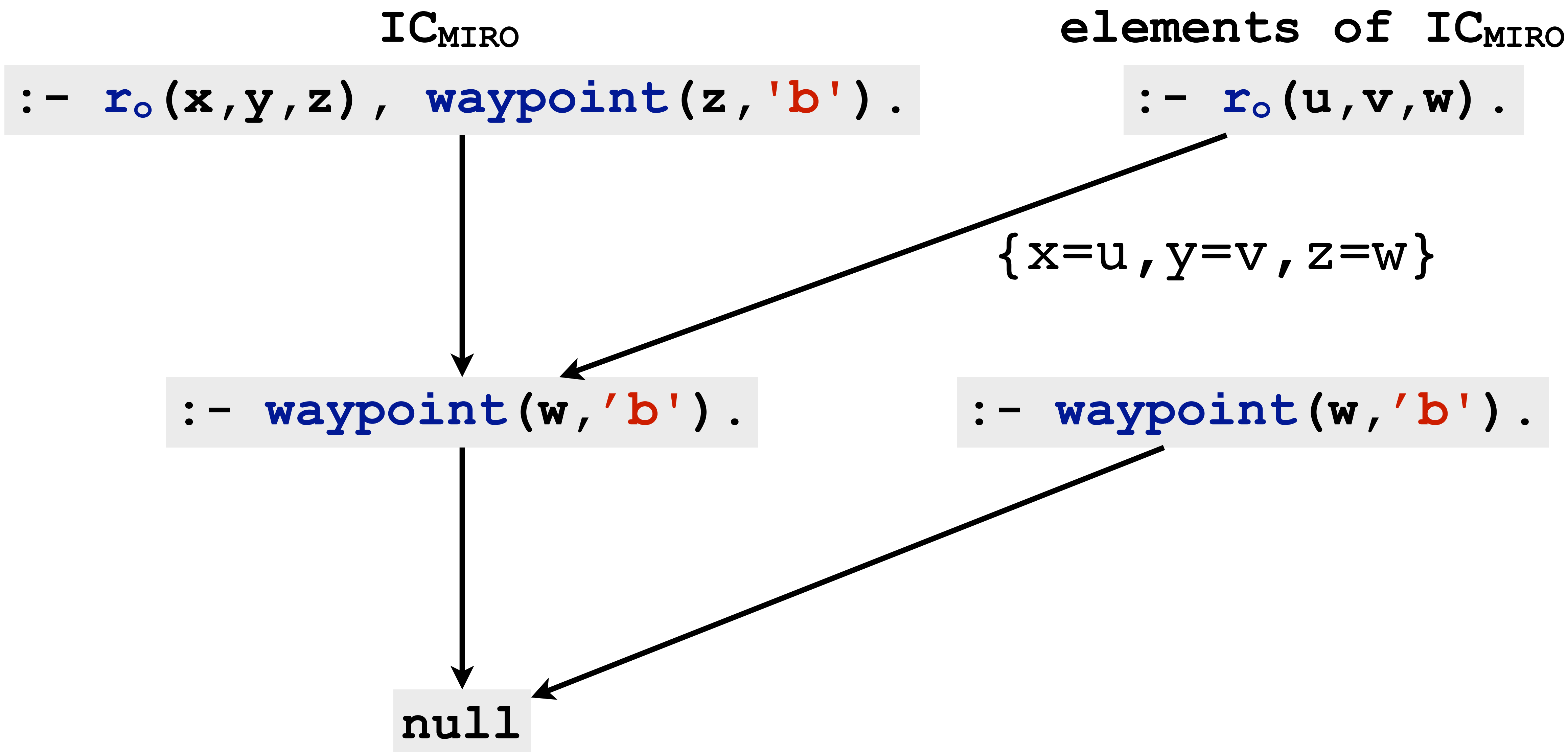
P_1 partially subsumes P_2

- if a subclass of P_1 subsumes P_2, signals policy interaction
- a fragment of P_1 — residue — that actually interacts with P_2 can be computed by the subsumption algorithm

the impact of P_1 on P2 is anticipated by the residue

residue is “null” — IC_{MIRO}, $IC_{MIRO'}$

IC_{MIRO} :- $r_\circ$(x,y,z), waypoint(z,'b').
$IC_{MIRO'}$:- $r_\circ$(u,v,w), waypoint(w,'b'), u = 'd'.



residue is “null” — IC_{MIRO}, $IC_{MIRO'}$

IC_{MIRO} :- $r_\circ$(x,y,z), waypoint(z,'b').
$IC_{MIRO'}$:- $r_\circ$(u,v,w), waypoint(w,'b'), u = 'd'.

IC_{MIRO}

:- $r_\circ$(x,y,z), waypoint(z,'b').

elements of $IC_{MIRO'}$

:- $r_\circ$(u,v,w).

{x=u,y=v,z=w}

:- waypoint(w,'b').

:- waypoint(w,'b').

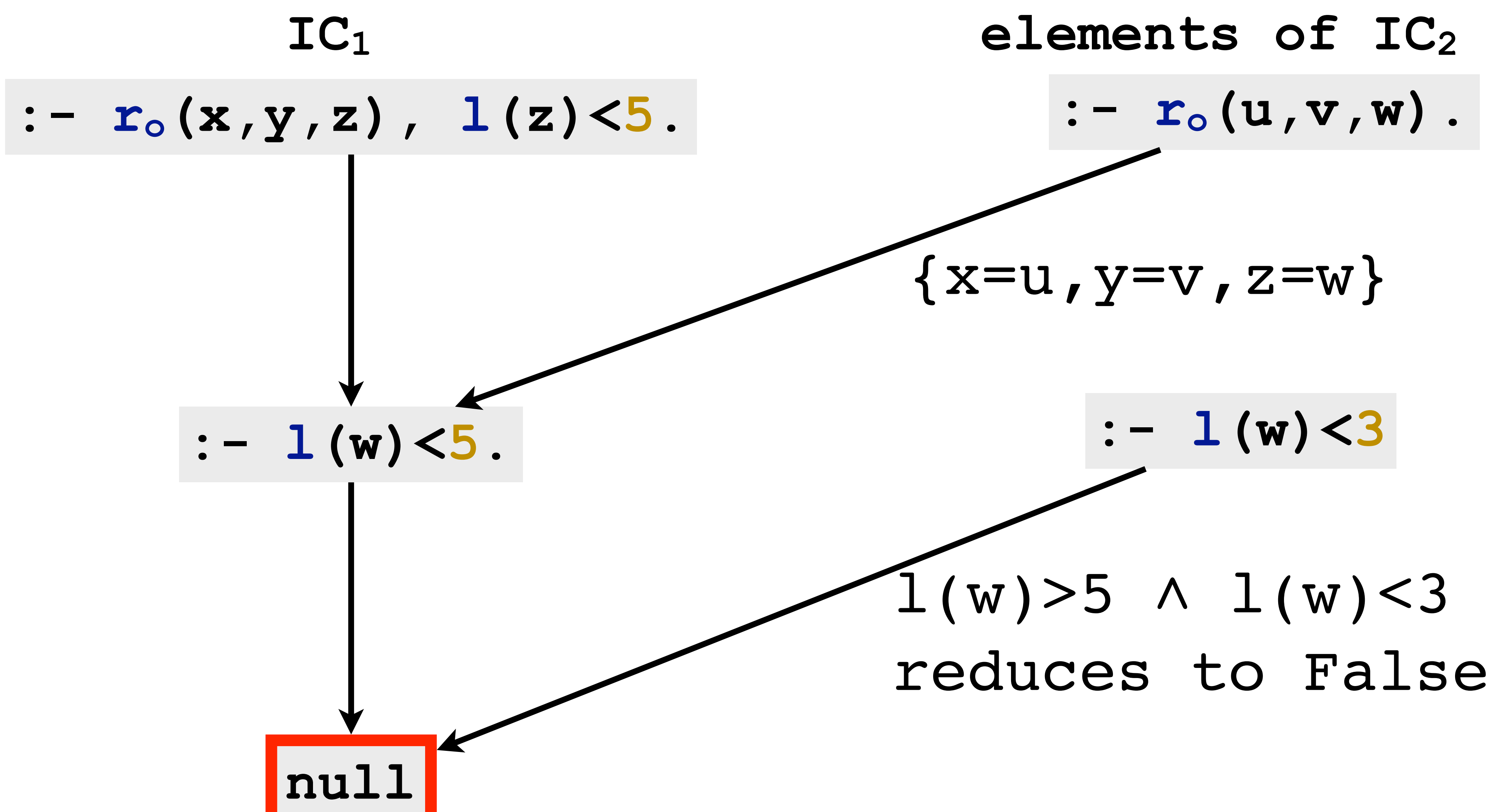
null

a “null” residue shows IC_{MIRO} is stronger (subsumption test succeeds)

IC_{MIRO} is stronger than $IC_{MIRO'}$

residue is "null" — IC_1, IC_2

```
IC₁:  :- r∘(x,y,z), l(z)<5.
IC₂:  :- r∘(u,v,w), l(w)<3, w=['AS2','AS3'].
```



IC_1 is stronger than IC_2

residue is "trivial" — IC_3, IC_4

disjoint

IC_3: `:- r∘(x,y,z),` $\boxed{\texttt{cust(x),}}$ `l(z)<5.`

IC_4: `:- r∘(u,v,w),` $\boxed{\texttt{admin(u)}}$`, waypoint(z,'b').`

IC_1	elements of IC_2
`:- r∘(x,y,z), cust(x), l(z)<5.`	no resolution possible

IC_3 is independent of IC_4

residue is non-trivial — IC_{sp}, IC_{ep}

%% shortest path

IC_{sp} :- $r_o(x,y,z)$, $r_i(x,y_2,z_2)$, $l(z)>l(z_2)$.

%% explicit path policy

IC_{ep}: z=a :- $r_o(x,y,z)$, x=d.

$$\begin{array}{ccc} IC_{ep} & & \text{elements of } IC_{sp} \\ \texttt{z=a :- } r_o(x,y,z)\texttt{, x=d.} & & \texttt{:- } r_o(x,y,z)\texttt{.} \\ \downarrow & \swarrow & \\ \texttt{z=a :- x=d.} & & \end{array}$$

residue is non-trivial — IC_{sp}, IC_{ep}

```
%% shortest path
ICsp :- ro(x,y,z), ri(x,y2,z2), l(z)>l(z2).
%% explicit path policy
ICep: z=a :- ro(x,y,z), x=d.
```

IC_{ep} — `z=a :- ro(x,y,z), x=d.`

elements of IC_{sp} — `:- ro(x,y,z).`

$$\boxed{\texttt{z=a :- x=d.}}$$

a non-trivial residue prescribes the impact of IC_{ep} —
additional conditions that must be taken into account for IC_{sp}

IC_{sp} affects IC_{ep}, as anticipated by the residue

residue is non-trivial — IC_{sp}, IC_{ep}

%% *shortest path*

IC_{sp} :- $r_o(x,y,z)$, $r_i(x,y_2,z_2)$, $l(z)>l(z_2)$.

%% *explicit path policy*

IC_{ep}: z=a :- $r_o(x,y,z)$, x=d.

consider IC_{ep} more important — find shortest path only for destinations other than 'd'

IC_{ep}

z=a :- $r_o(x,y,z)$, x=d.

elements of IC_2

:- $r_o(x,y,z)$.

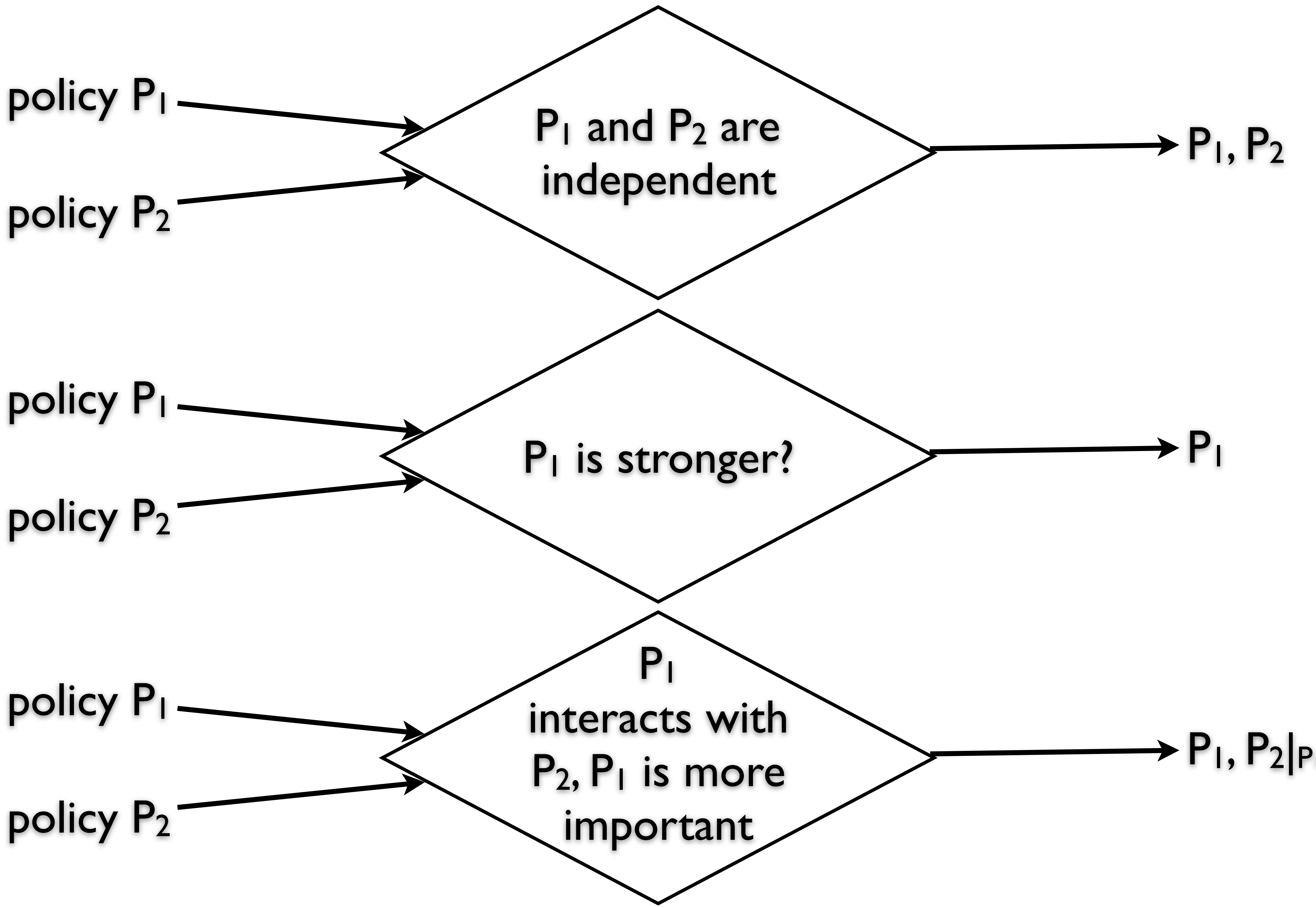
z=a :- x=d.

rewrite shortest path policy — semantically constrained with explicit path!

IC_{sp} :- $r_o(x,y,z)$, $r_i(x,y_2,z_2)$, $l(z)>l(z_2)$, {z=a :- x=d}.

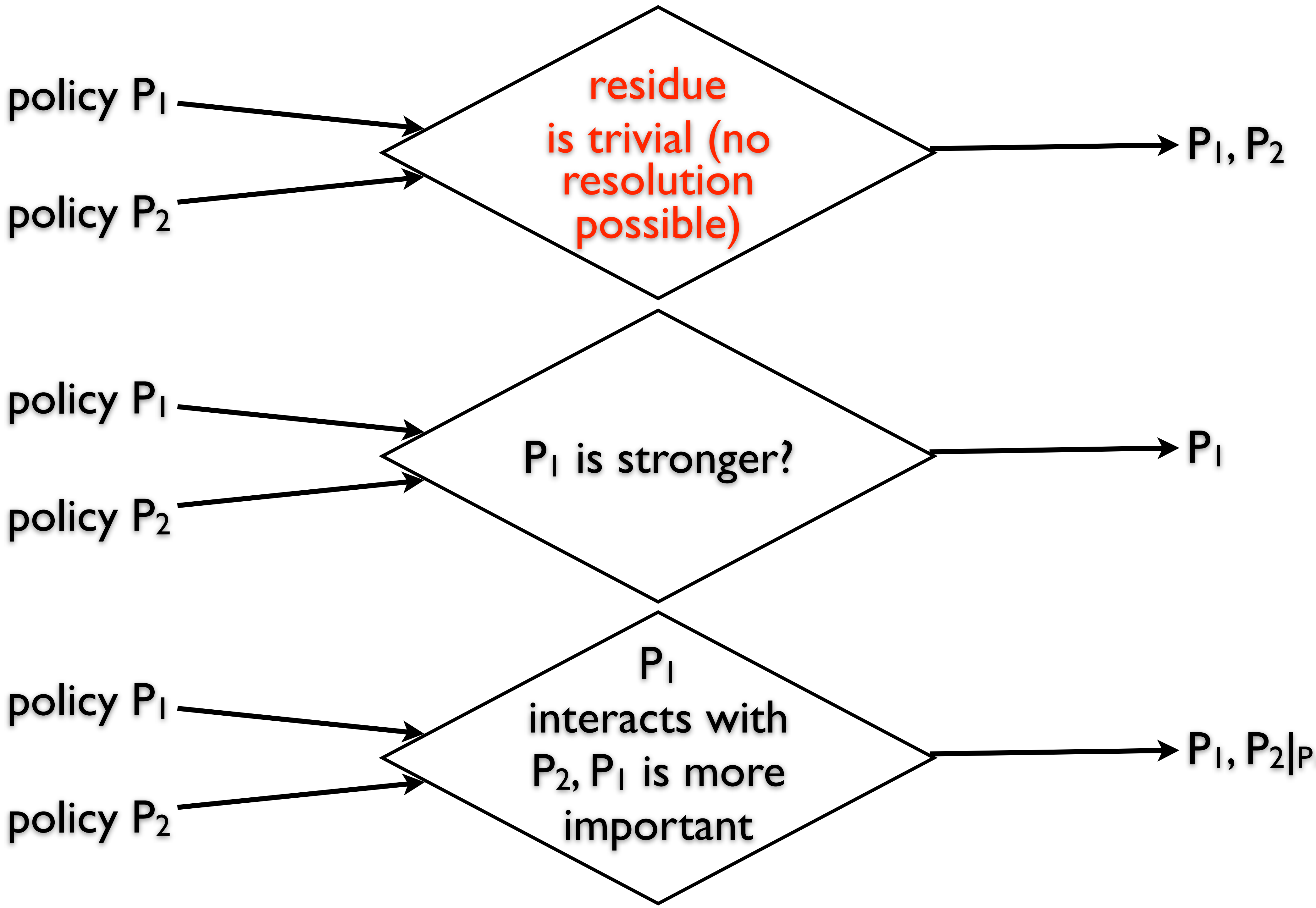
coordination by the residue method

residue — syntactic fragment that anticipates impact



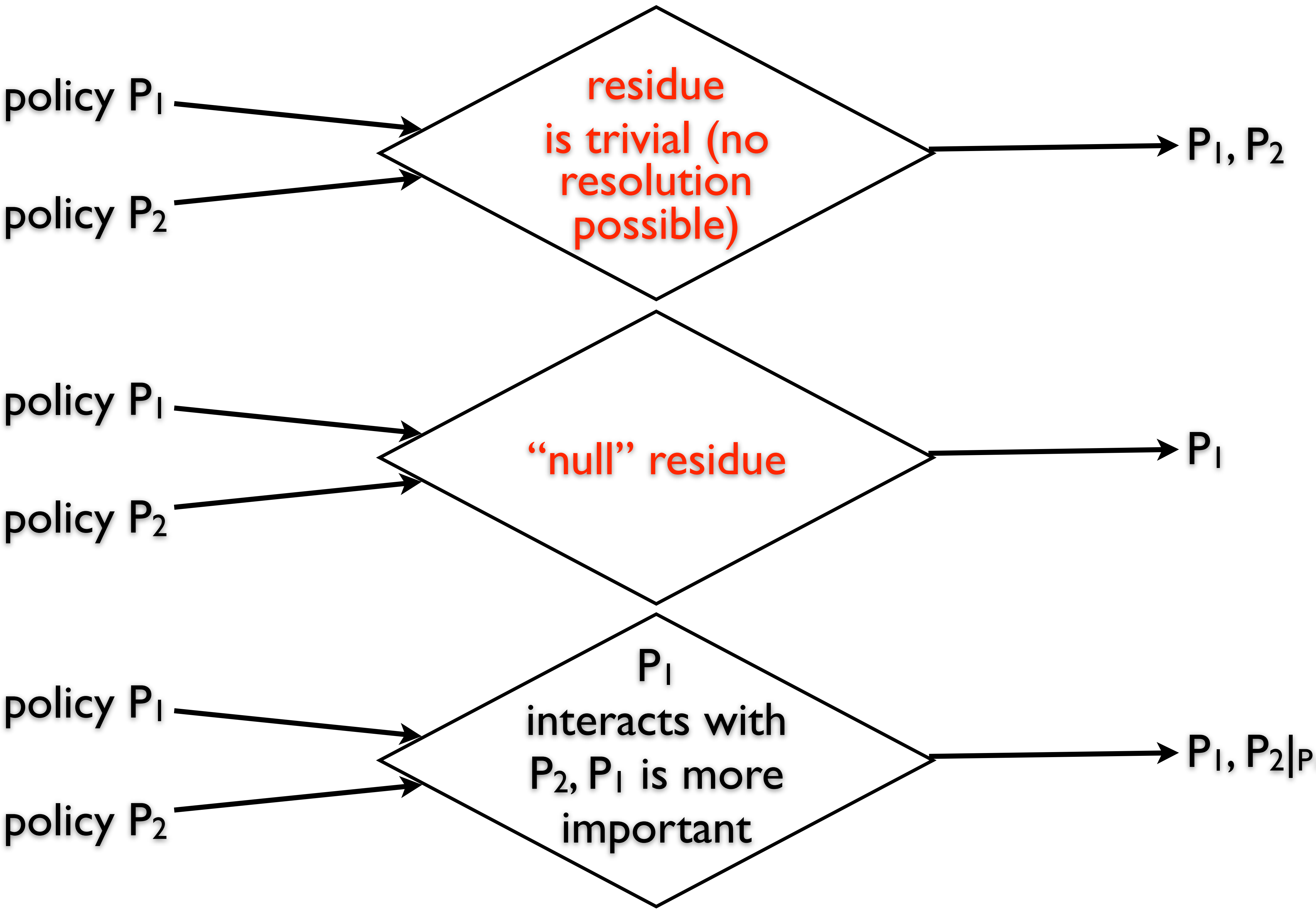
coordination by the residue method

residue — syntactic fragment that anticipates impact



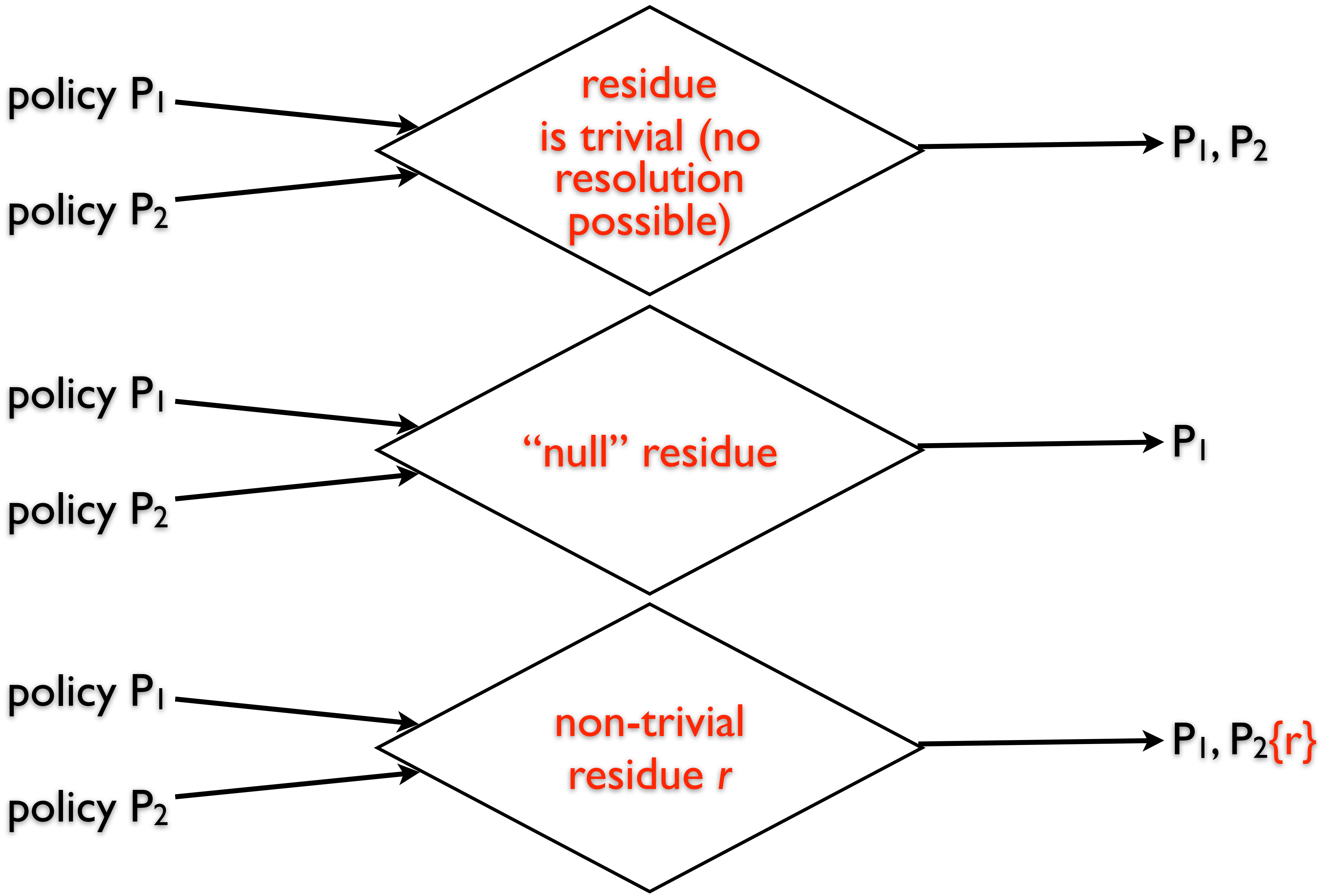
coordination by the residue method

residue — syntactic fragment that anticipates impact



coordination by the residue method

residue — syntactic fragment that anticipates impact



preliminary evaluation

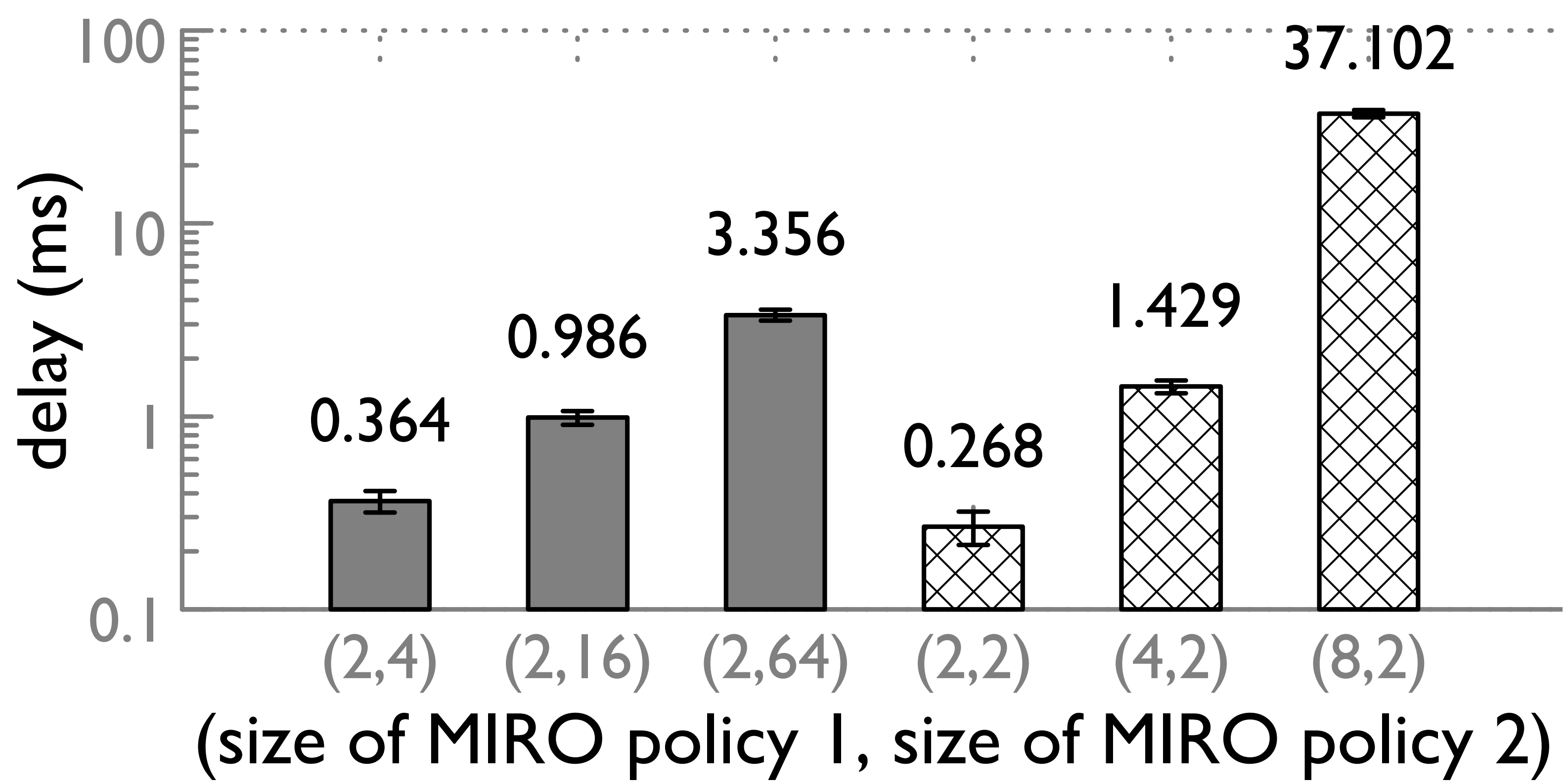
prototype

- implement the standard Θ-subsumption algorithm in Python
- macOS with 3.4GHz Intel Core i5 processor, 16GB RAM

preliminary evaluation

measure residue generation processing delay

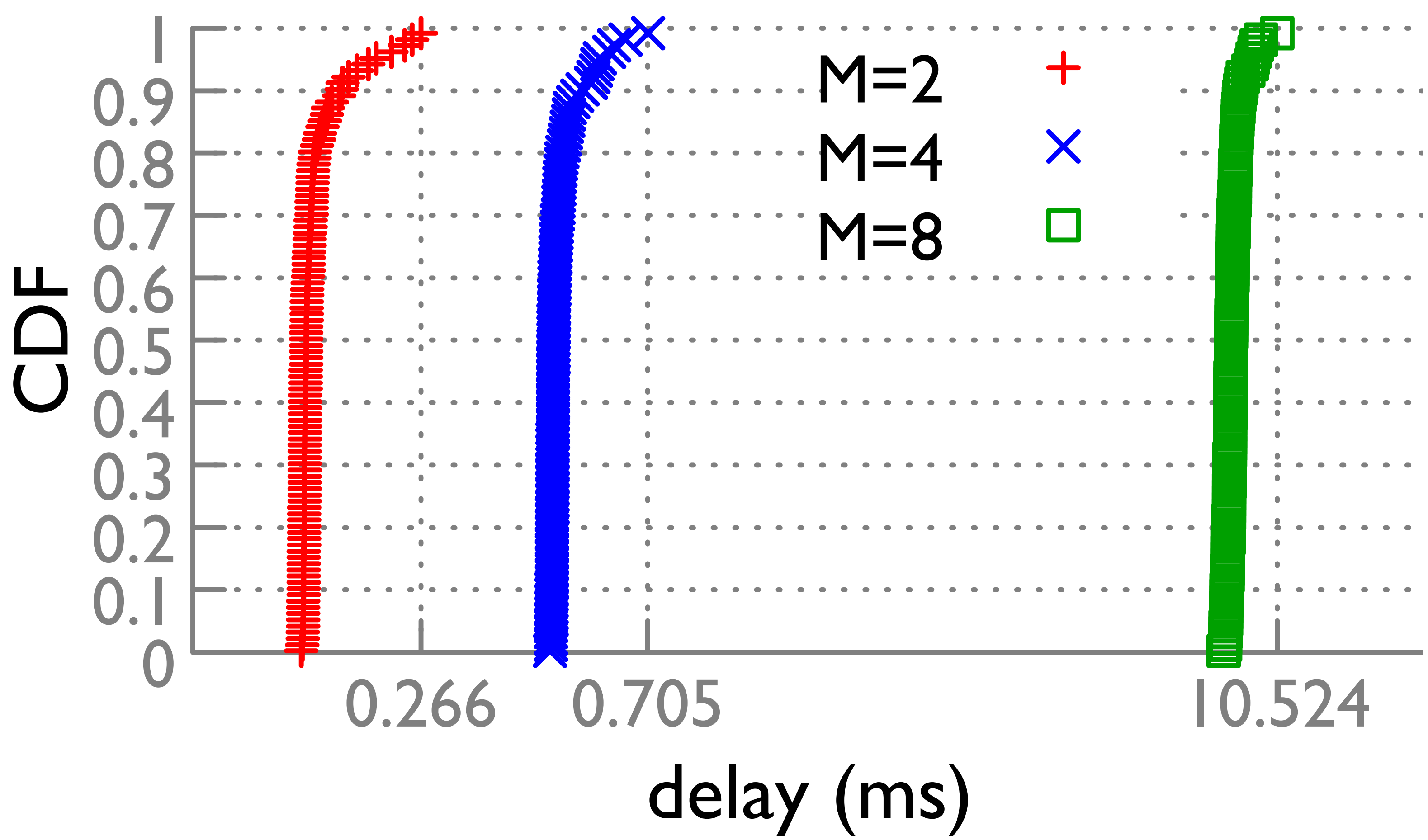
- two MIRO policies with varying sizes
  - policy size — number of randomly generated waypoint literals
- scale well
  - scale better when we fix the size of the subsuming policy



preliminary evaluation

measure residue generation processing delay

- MIRO policy of varying sizes (M)
- Wiser policy of 6 literals



summary

can we take a declarative approach towards Internet routing policies that are easier to manage?

Ravel: database-defined networking ravel-net.org