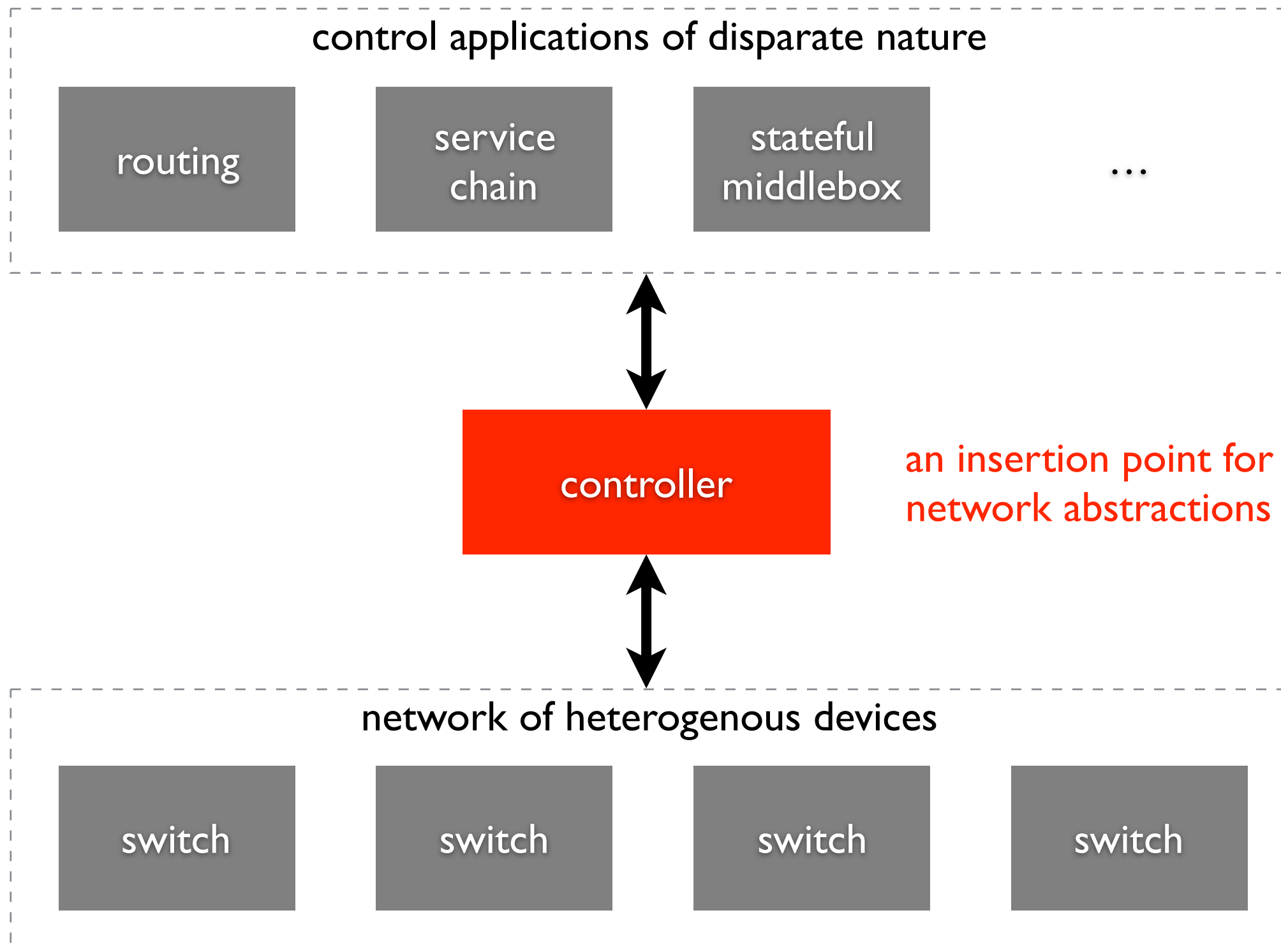




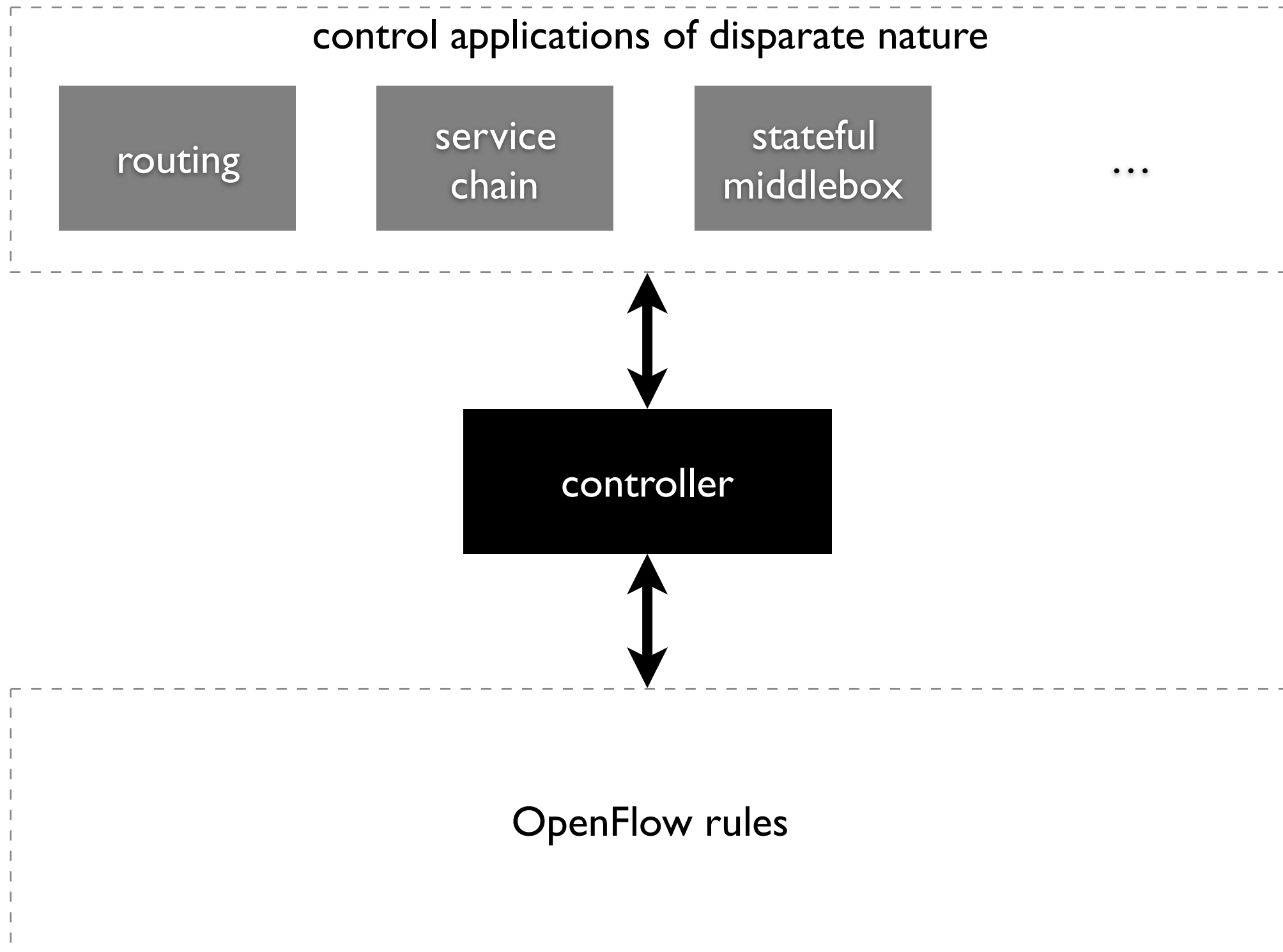
Ravel: a database-defined network

Anduo Wang Xueyuan Mei Jason Croft
Matthew Caesar Brighten Godfrey

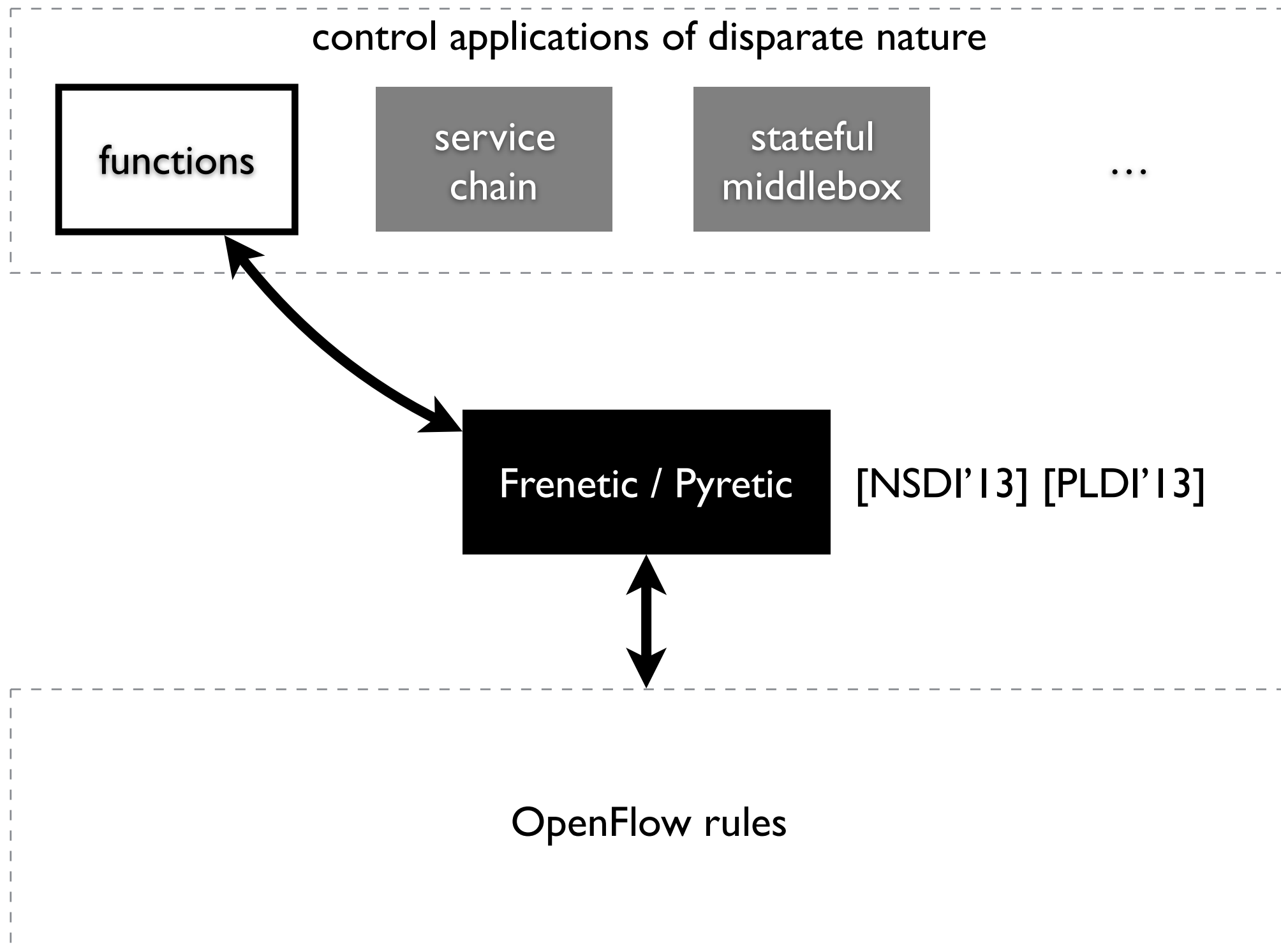
software-defined network



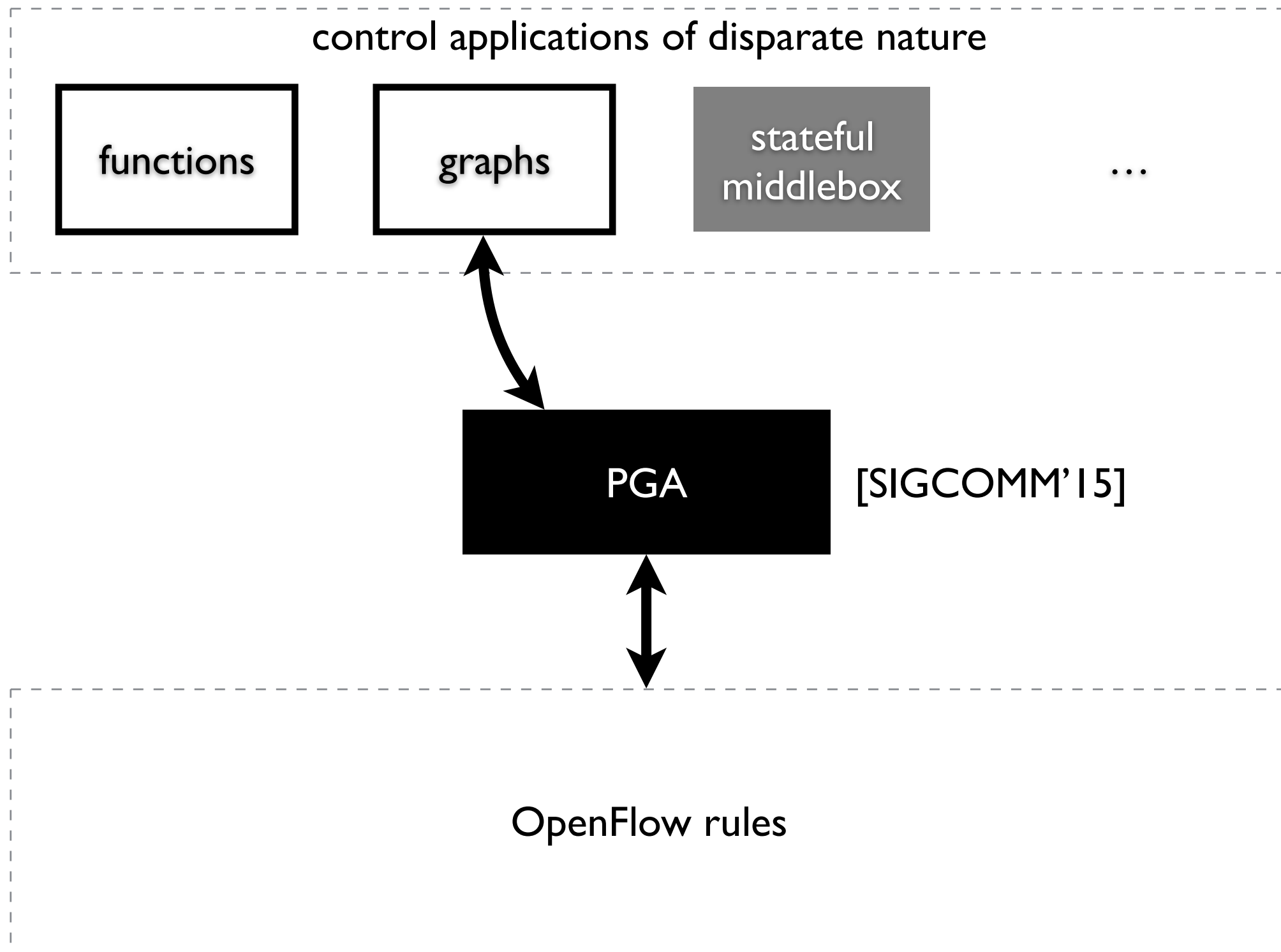
abstractions



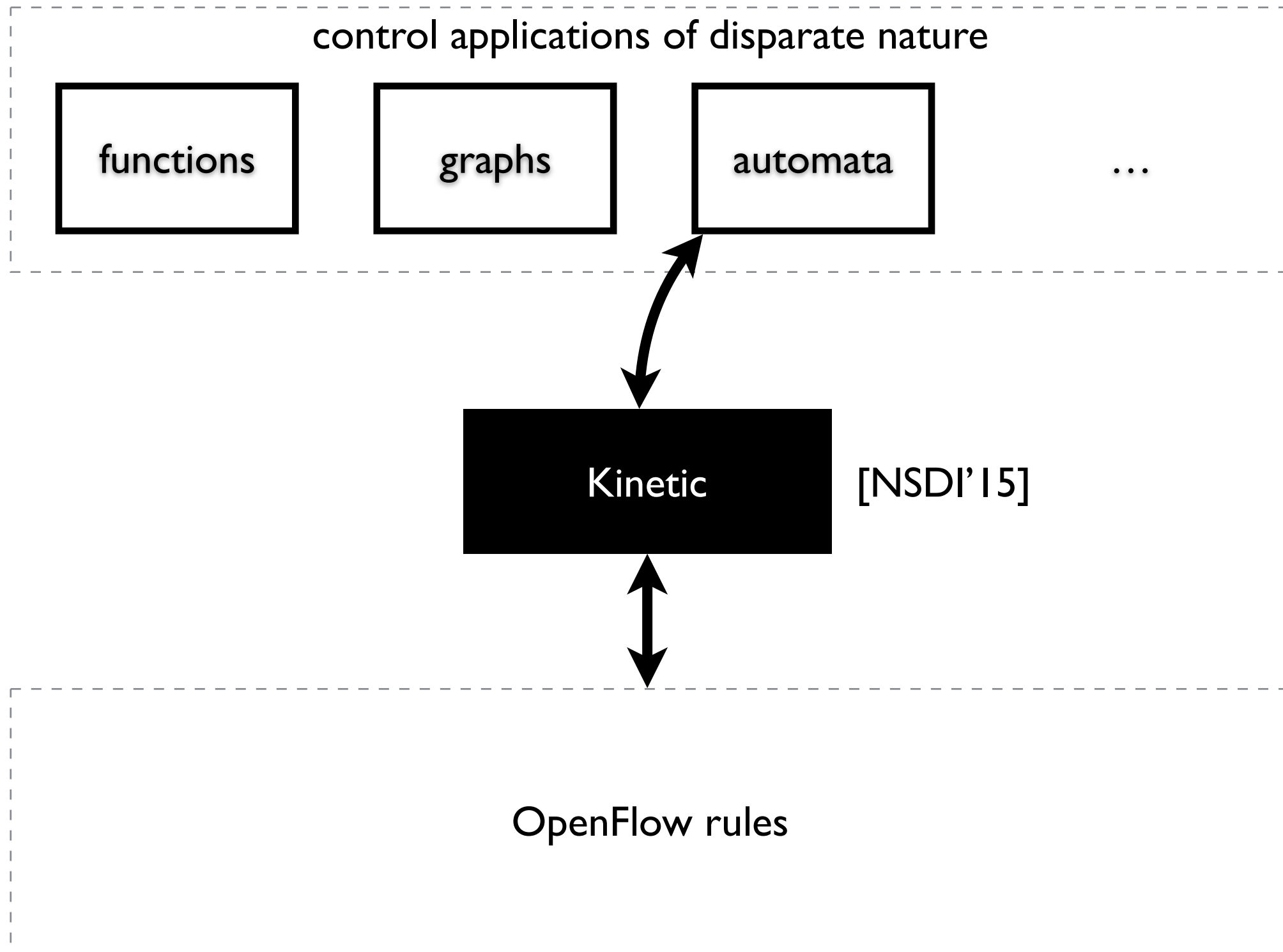
abstractions



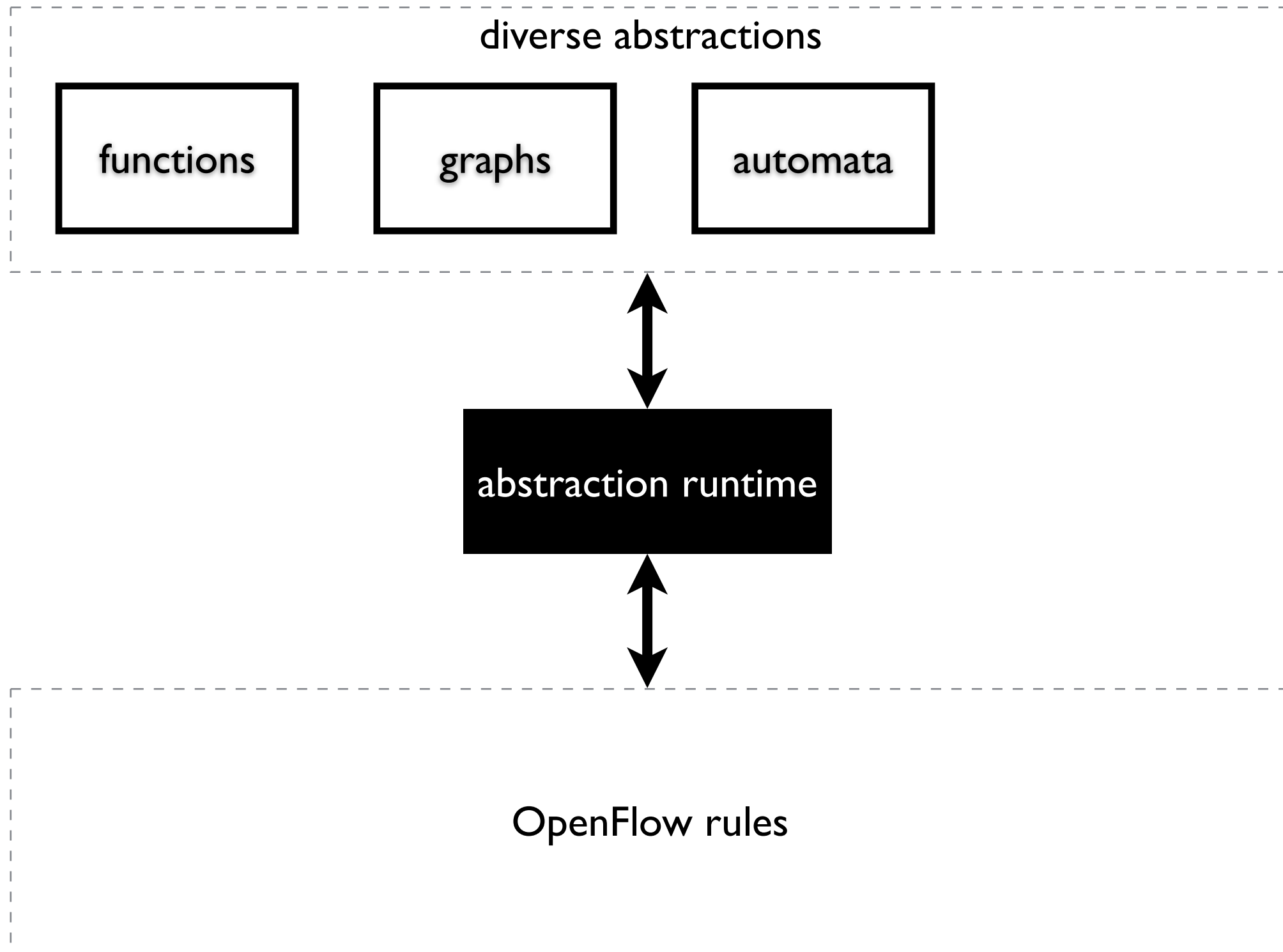
abstractions



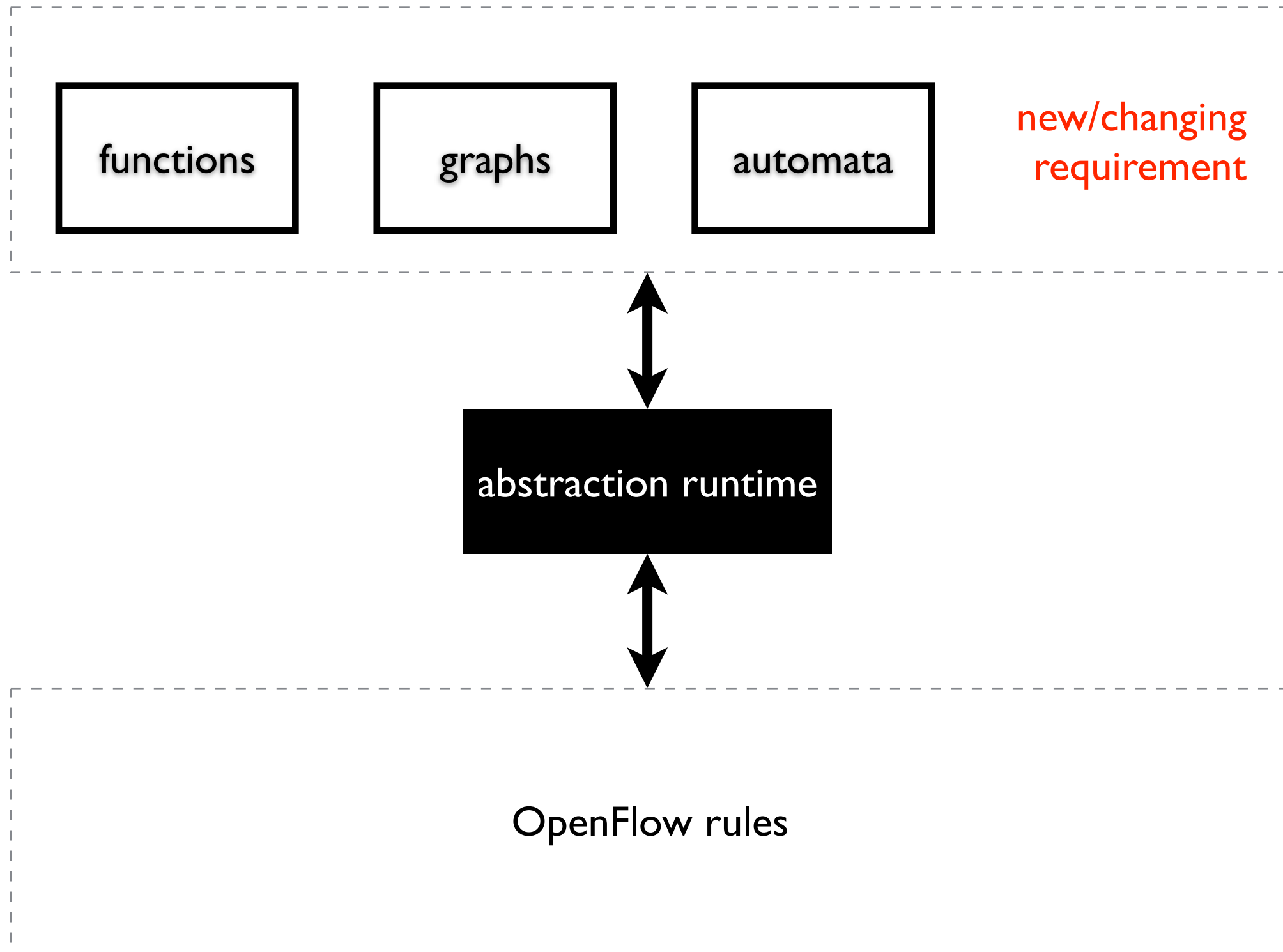
abstractions



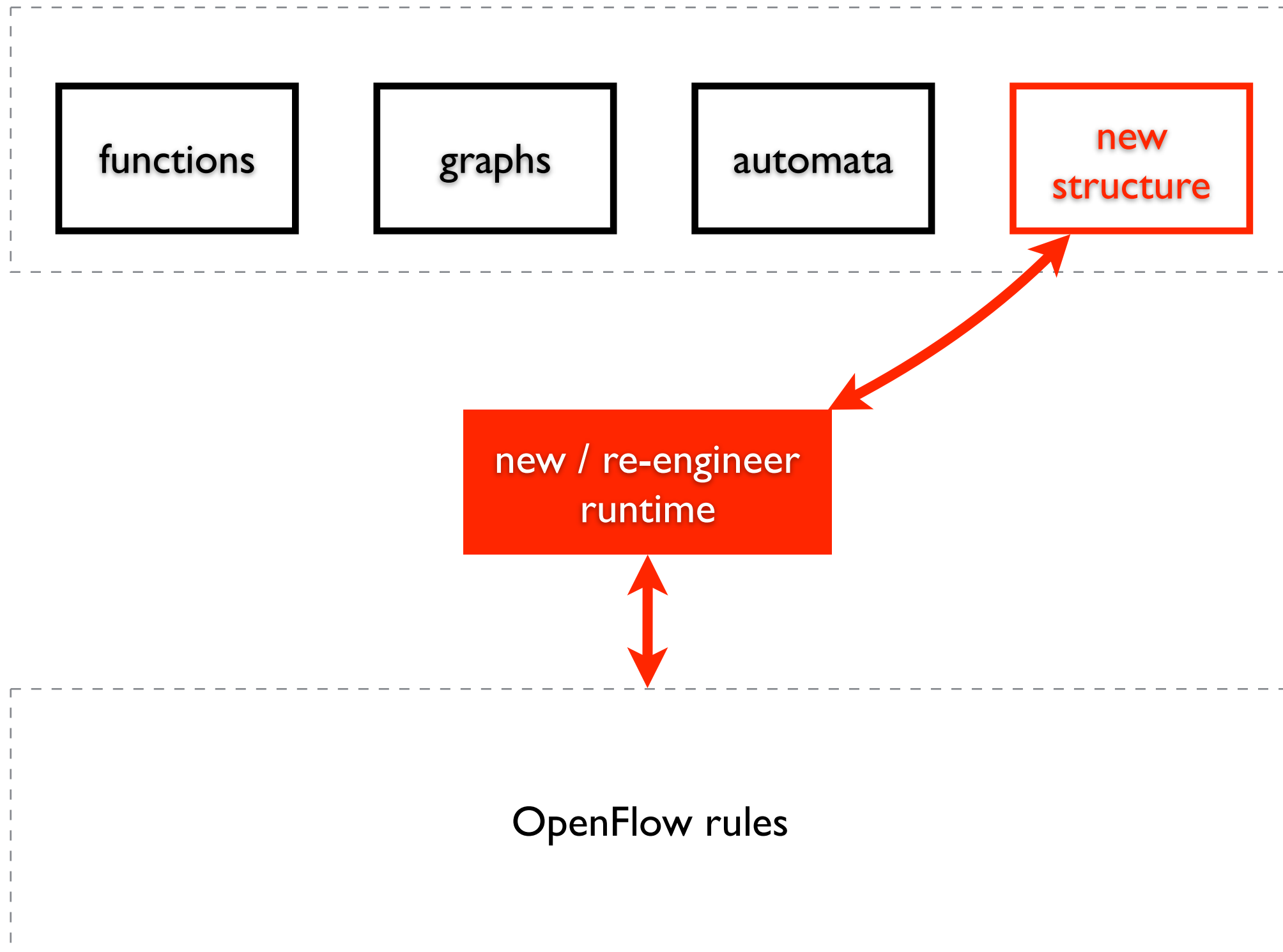
abstractions



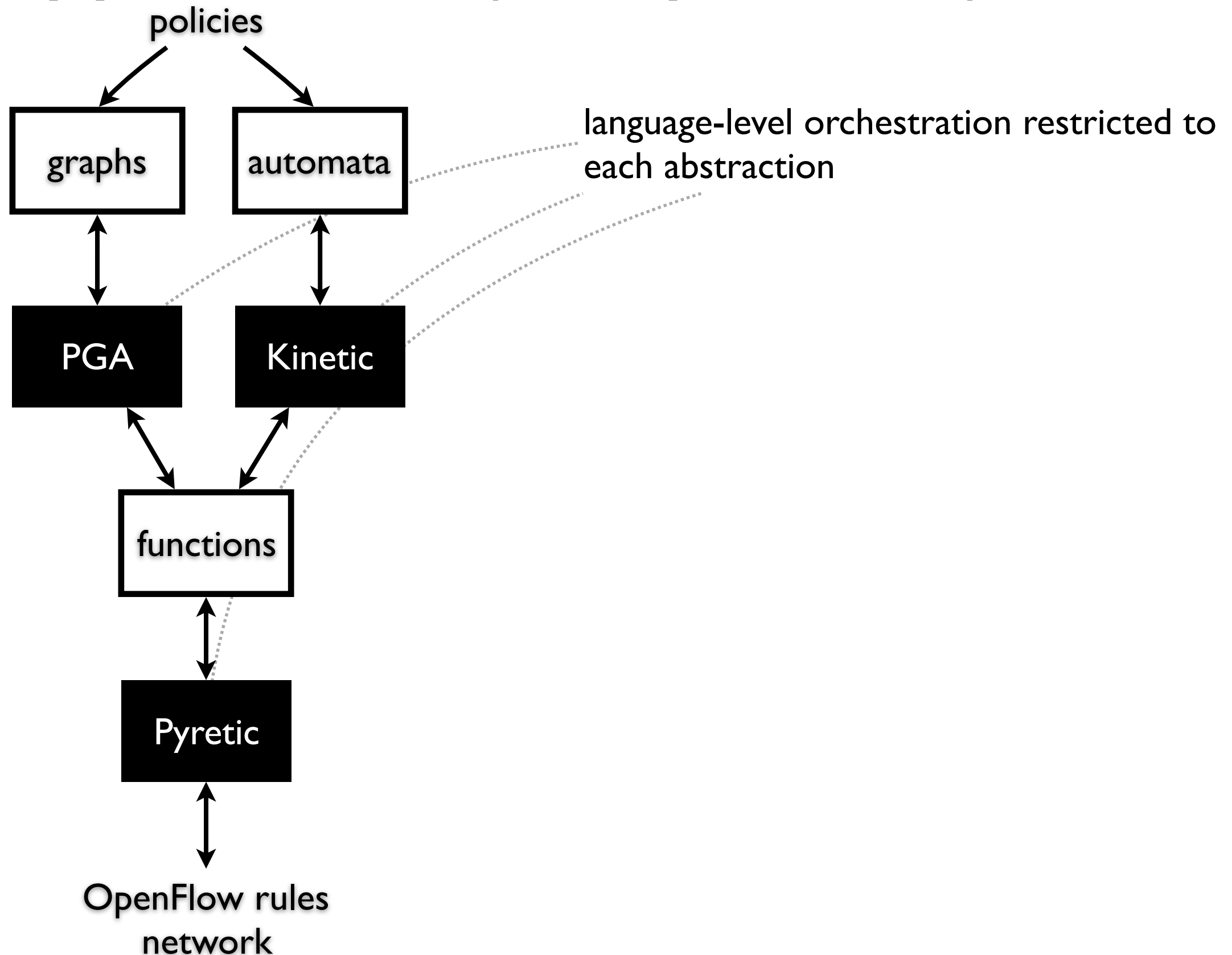
but network keep evolving



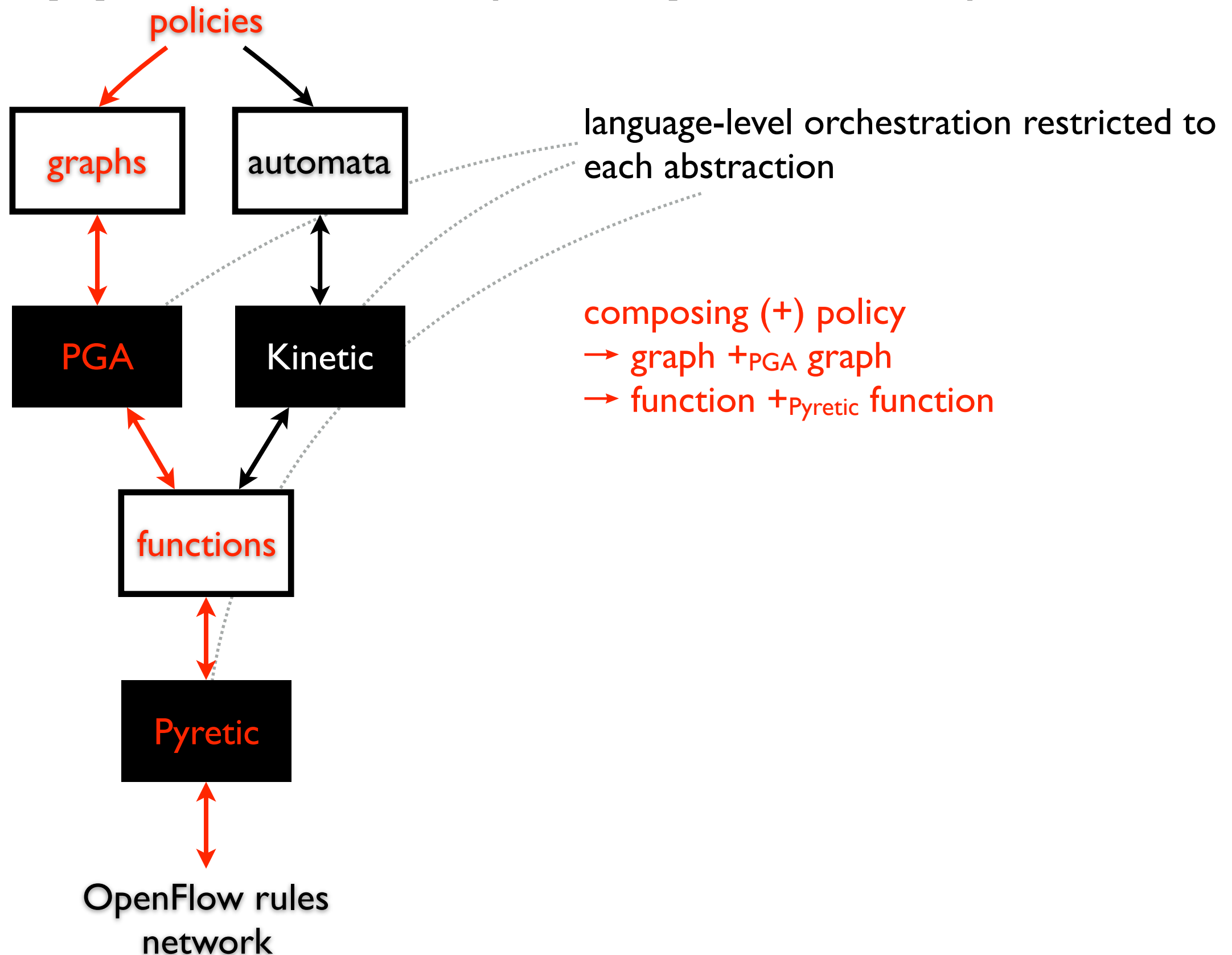
but network keep evolving



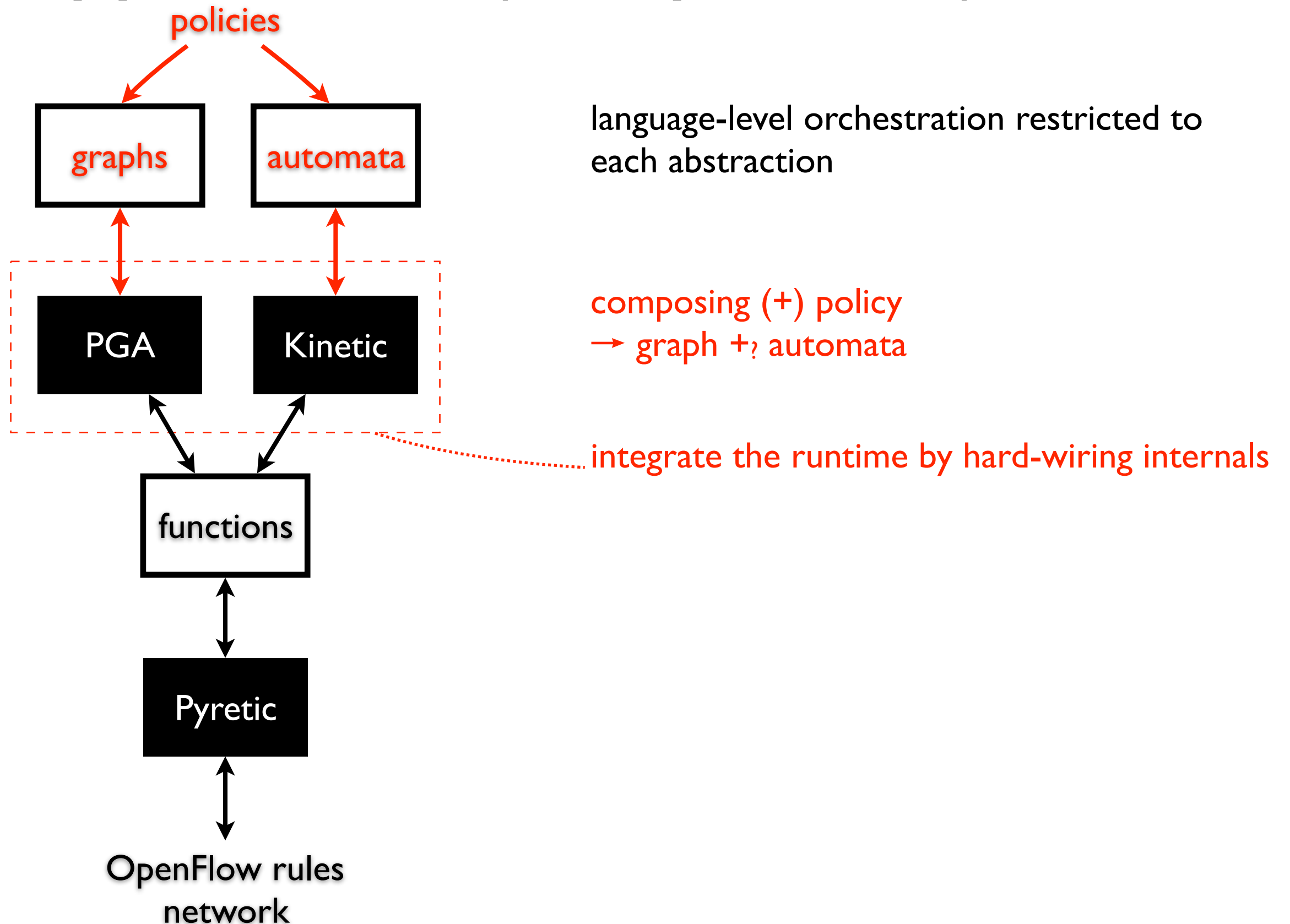
and applications (components) interact



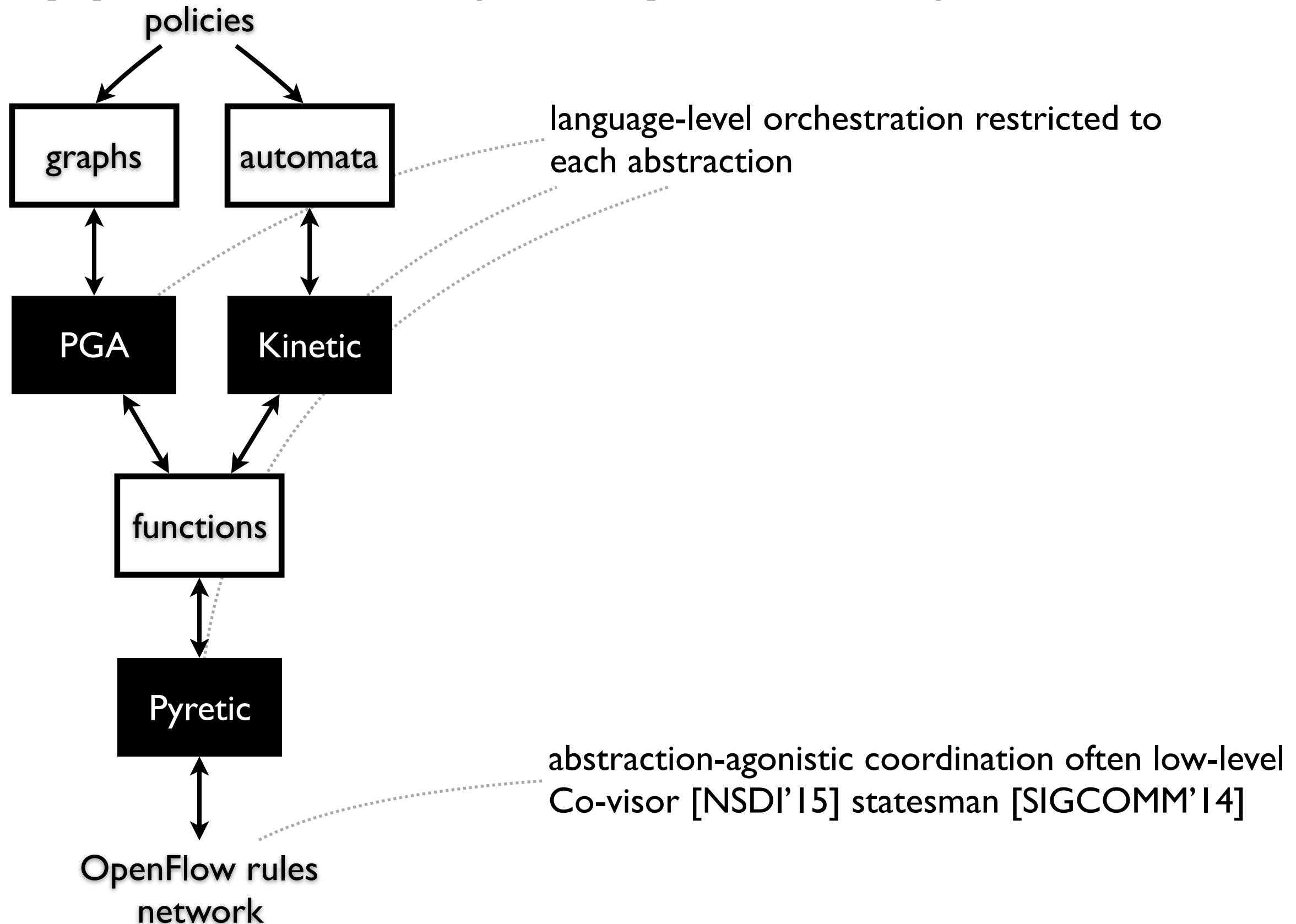
and applications (components) interact



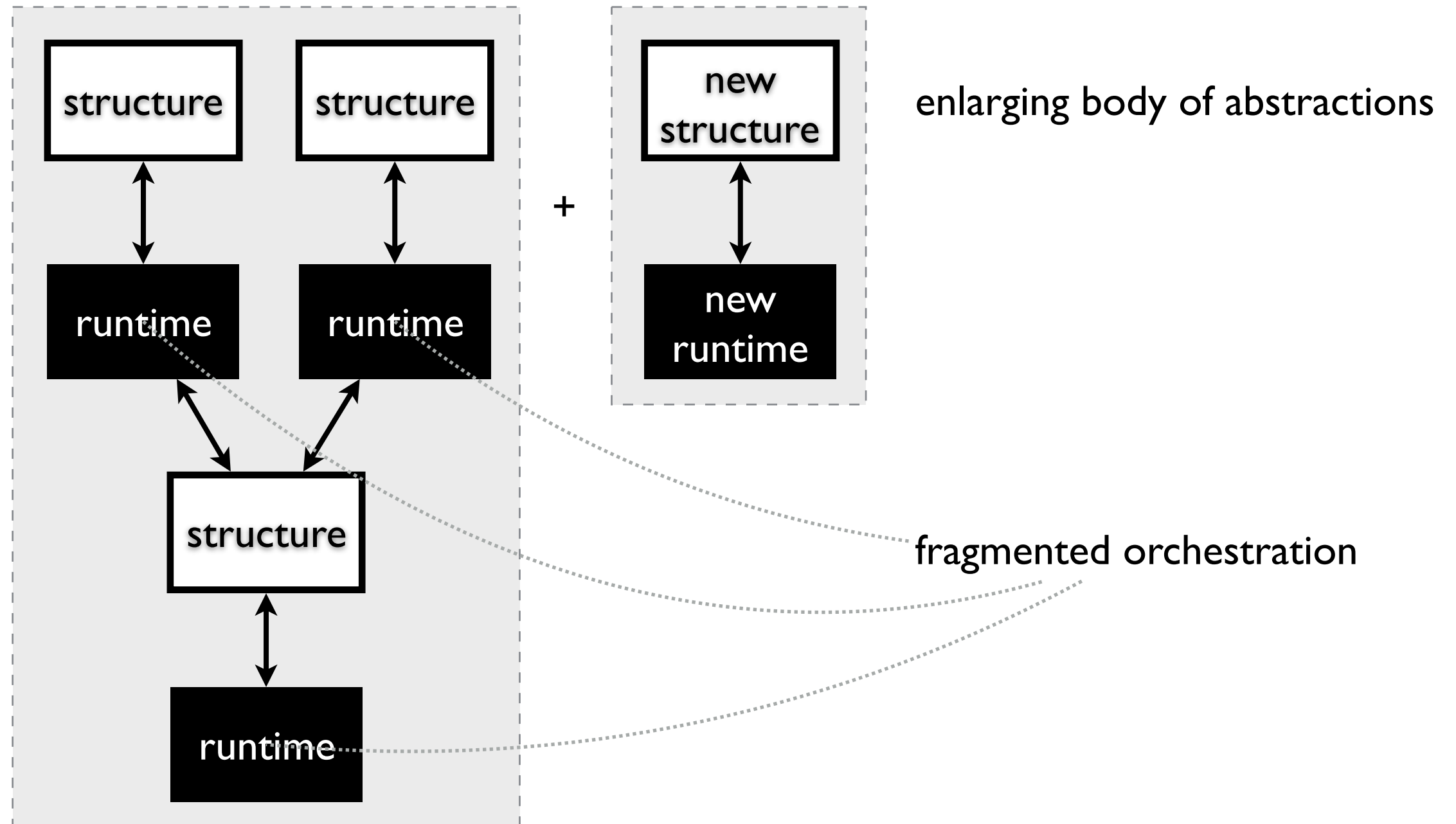
and applications (components) interact



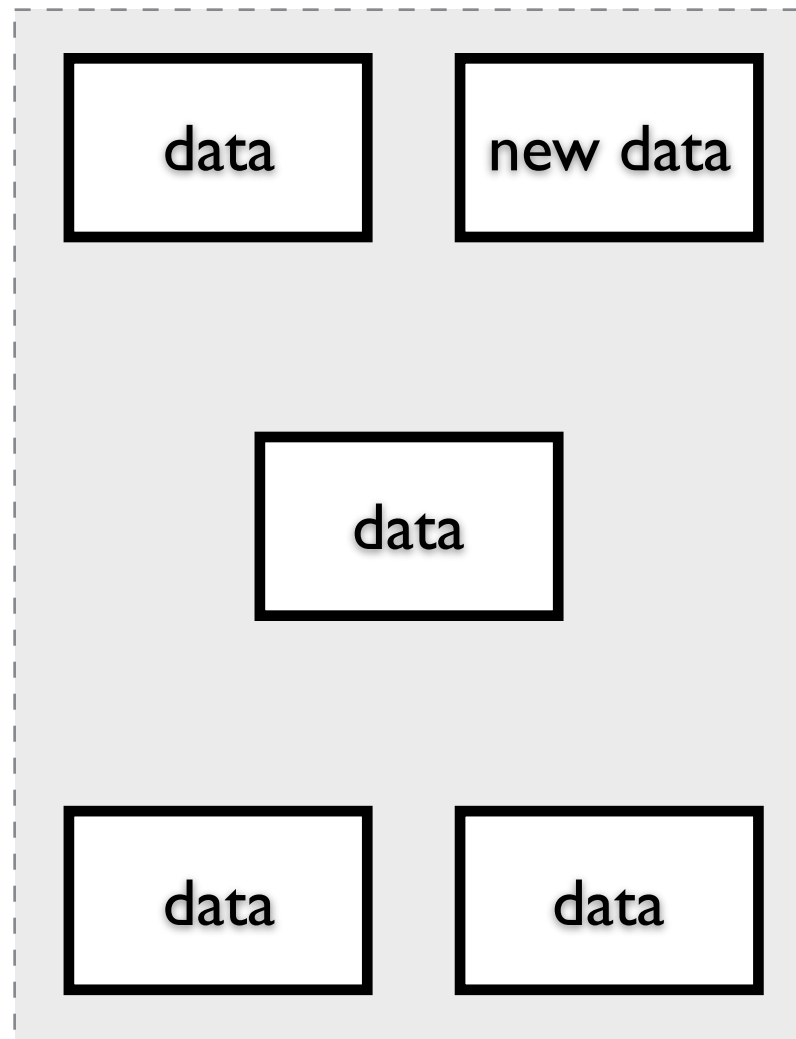
and applications (components) interact



current states of abstraction



our perspective



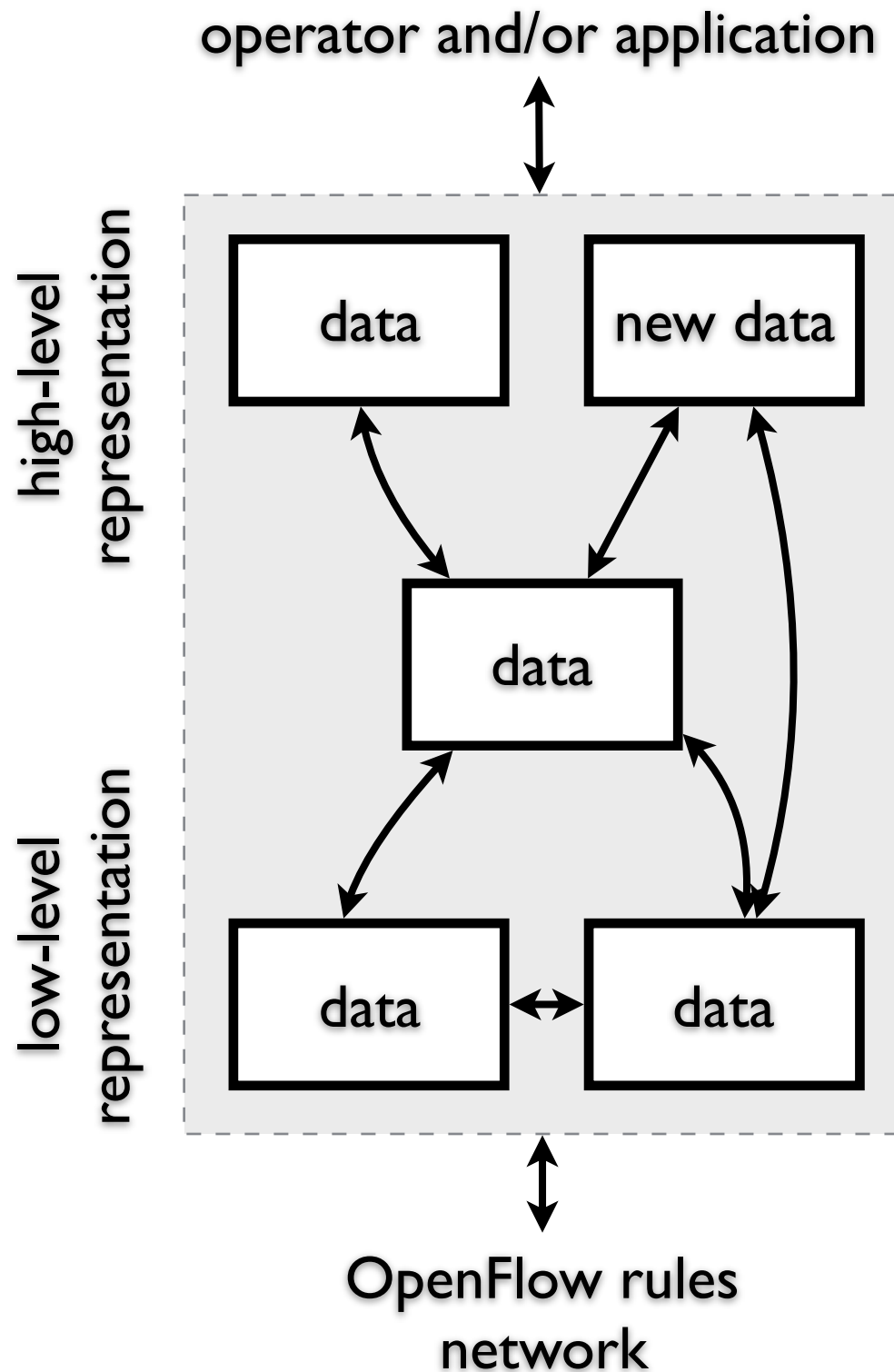
SDN control revolves around data representation

- discard specialized, pre-compiled, fixed structure
- adopt a *plain data representation*

our perspective

SDN control revolves around data representation

- discard specialized, pre-compiled, fixed structure
- adopt a *plain data representation*
- use a *universal data language*



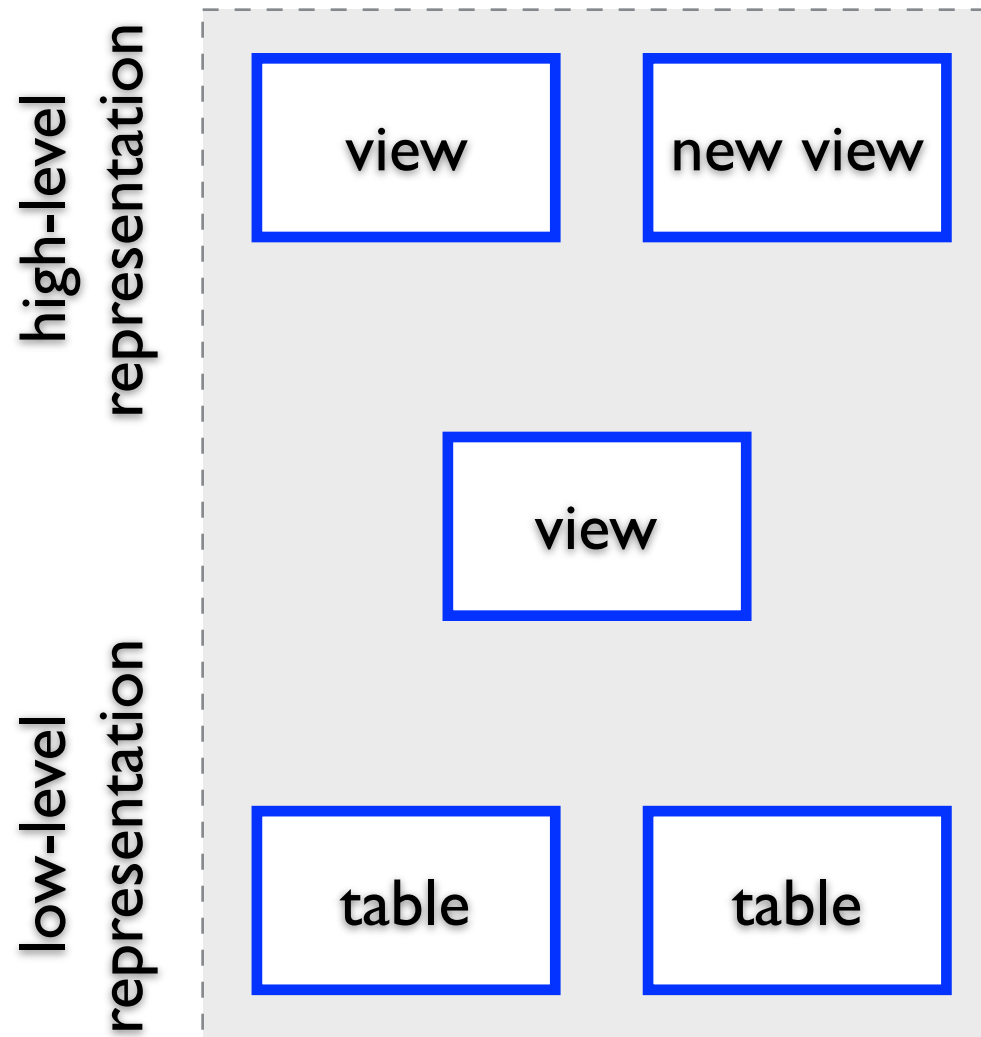
a database-defined network

operator and/or application

- **relation** — the plain data representation

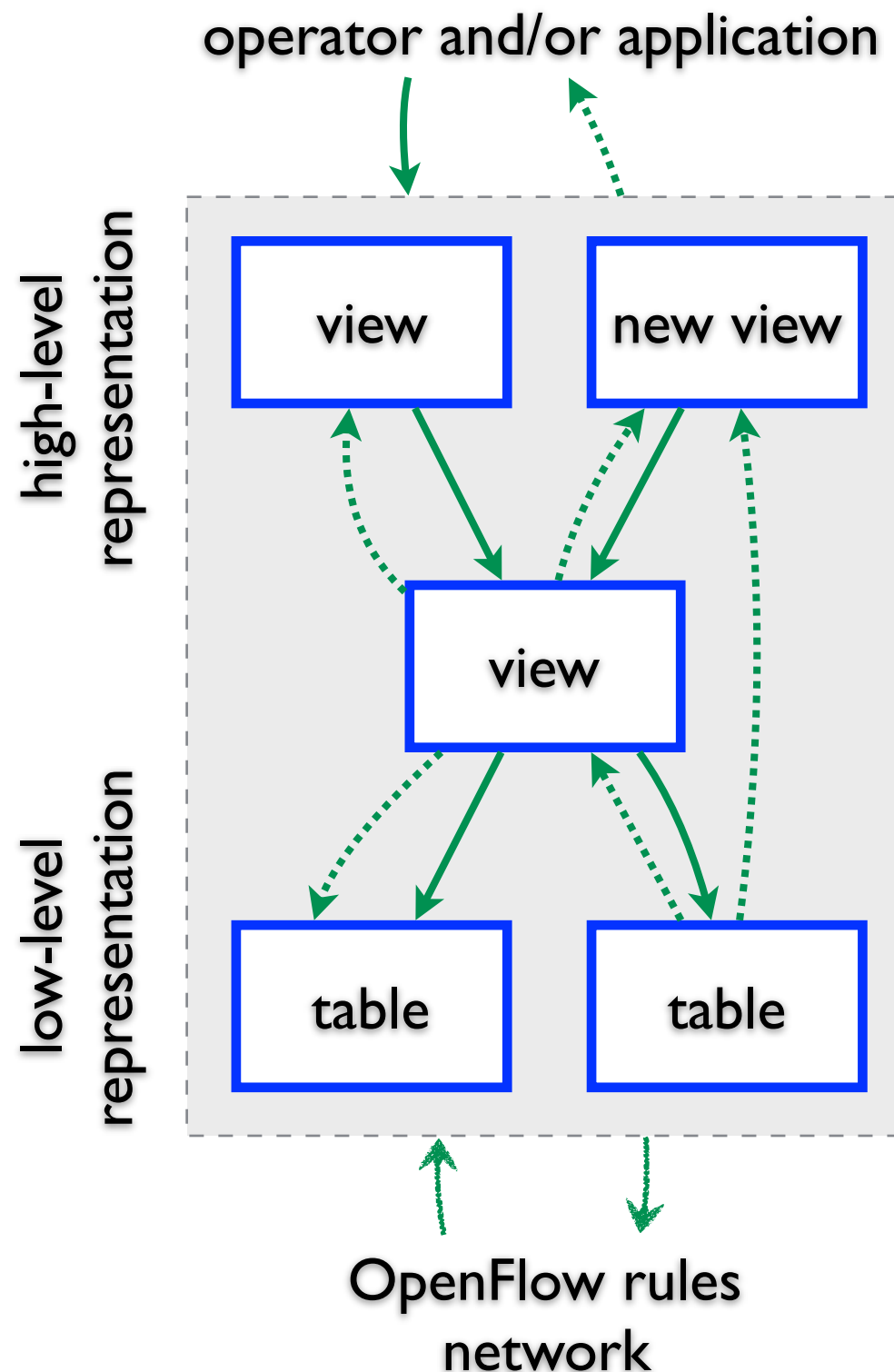
- table — stored relation

- view — virtual relation



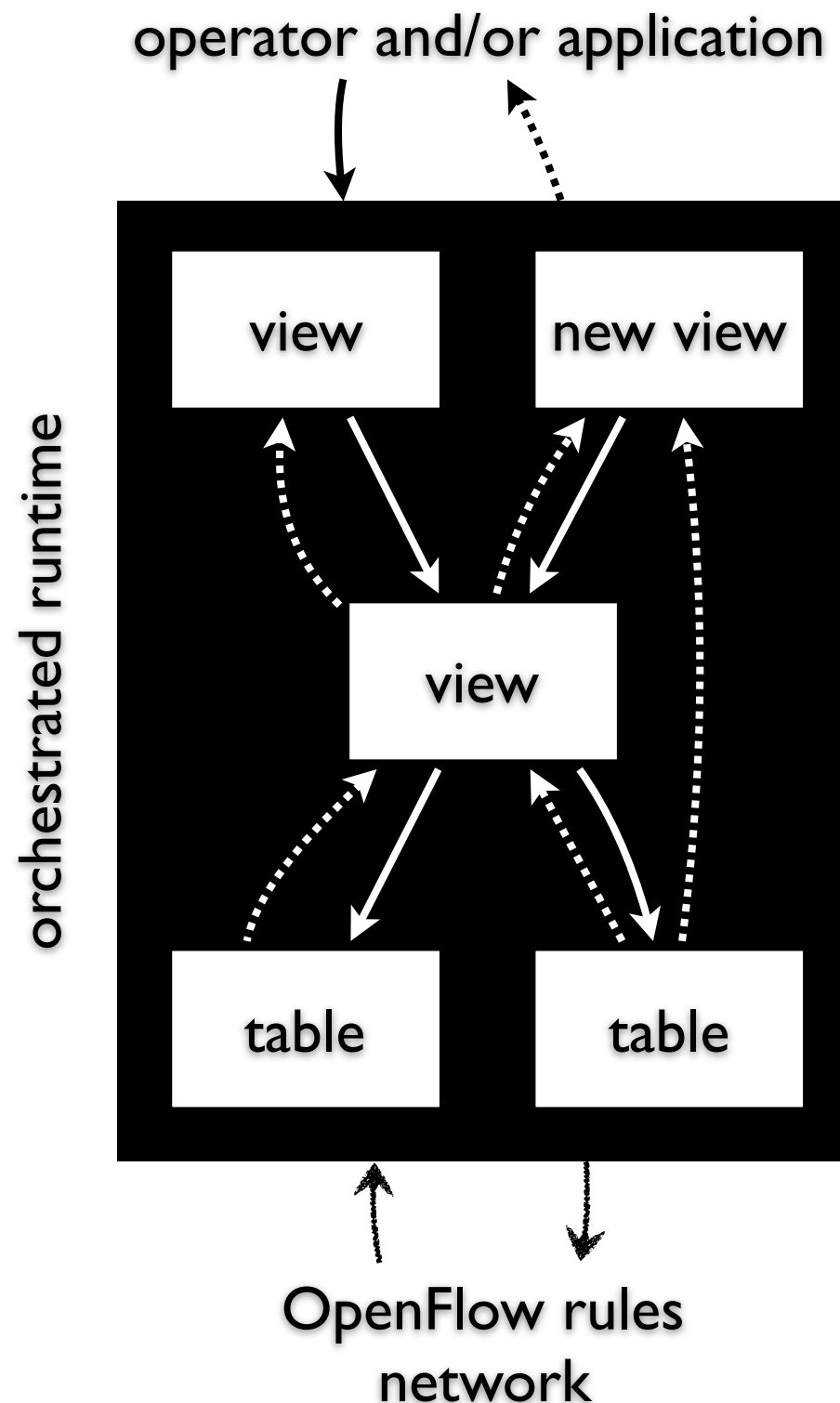
OpenFlow rules
network

a database-defined network



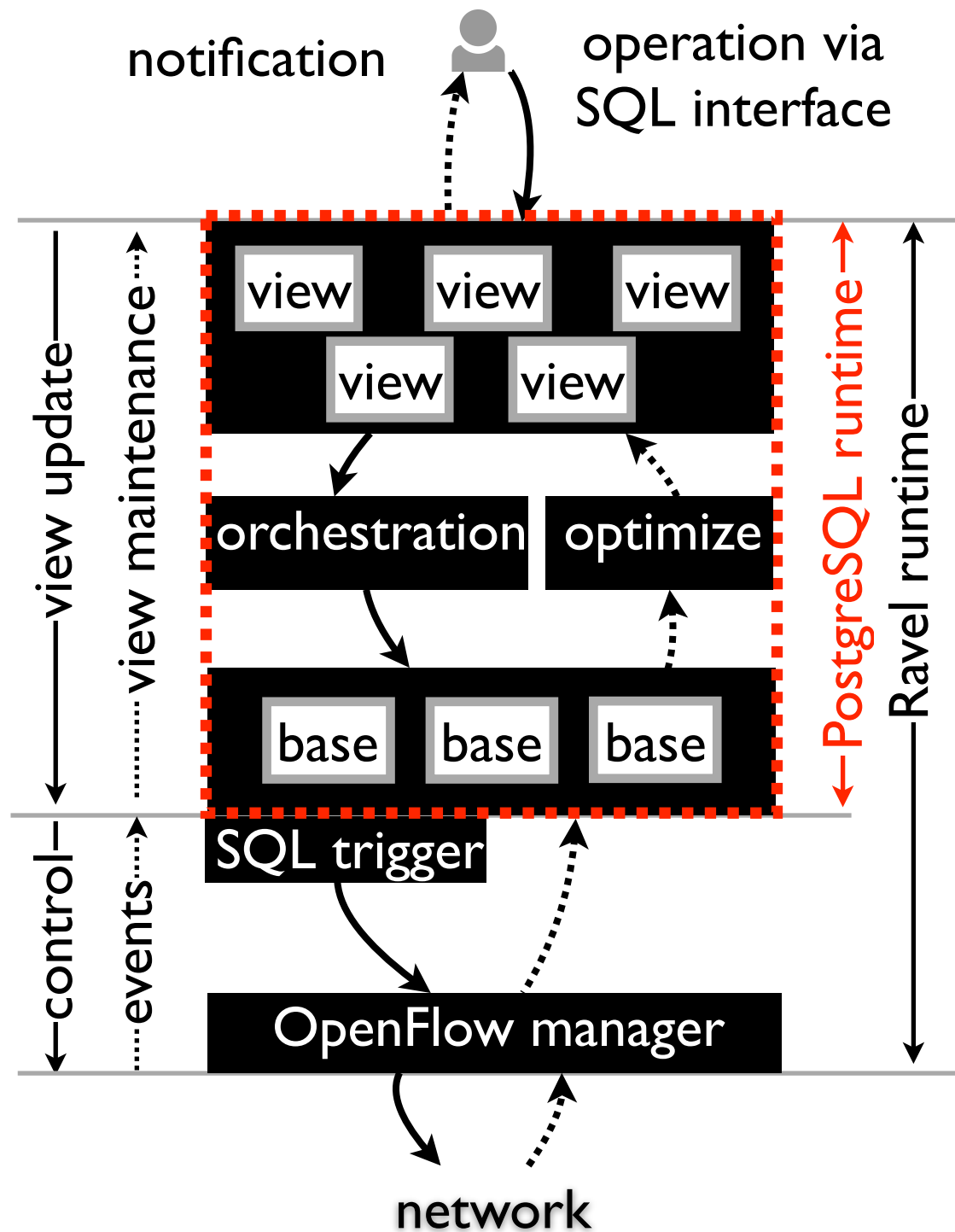
- **relation** — the plain data representation
 - table — stored relation
 - view — virtual relation
- **SQL** — the universal data language
 - SQL query
 - SQL update ———
 - SQL trigger ~~~~~

a database-defined network



- relation — the plain data representation
 - table — stored relation
 - view — virtual relation
- SQL — the universal data language
 - SQL query
 - SQL update ———
 - SQL trigger ~~~~~
- SQL database — the high-performance runtime
 - orchestration challenge: refine runtime behavior by data mediation

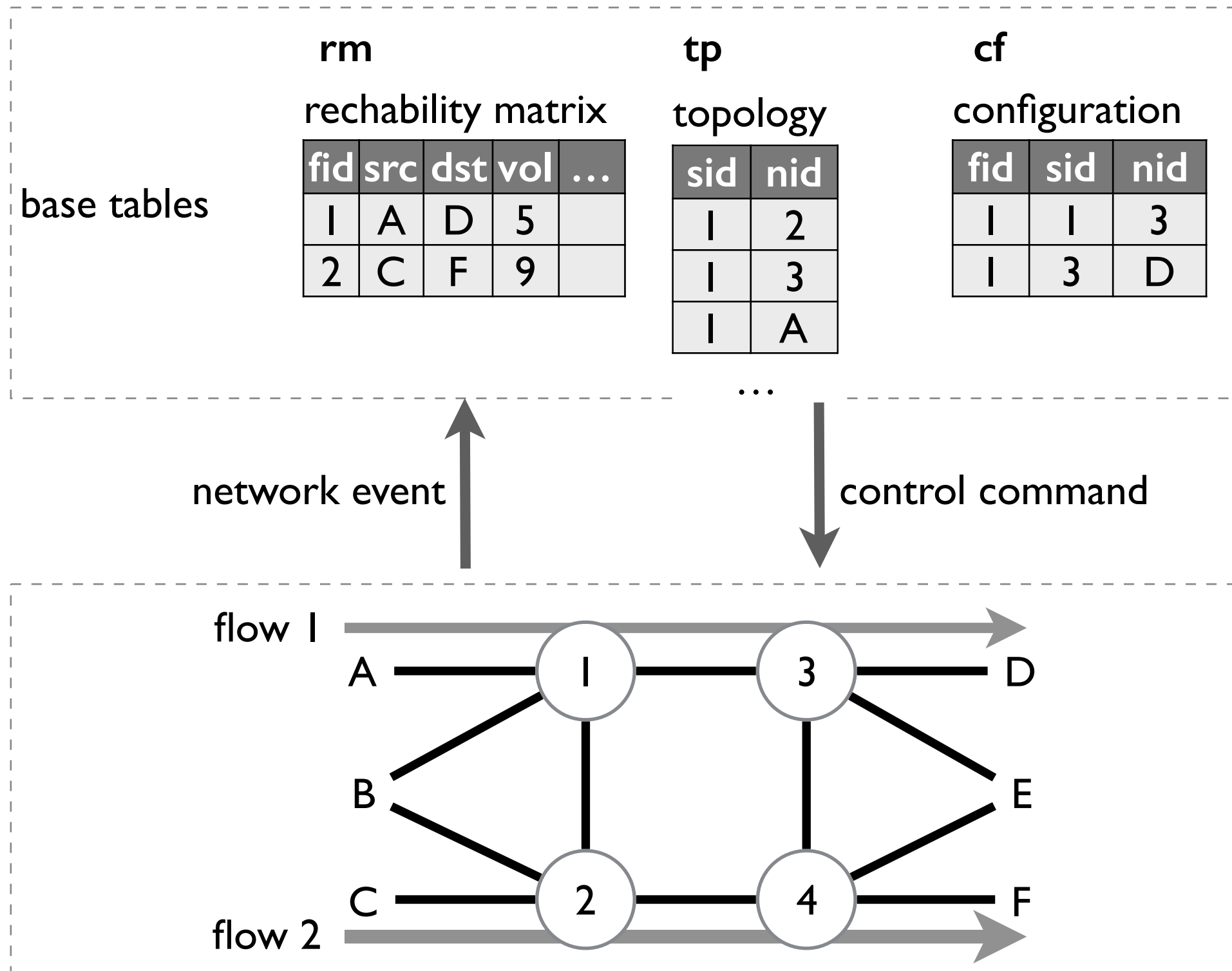
Ravel: a realization with SQL database



attractive features

- ad-hoc programmable abstraction via views
- orchestration across abstractions via view mechanism
- orchestration across applications via data mediation
- network control via SQL

abstraction: network tables



abstraction: application view

firewall-specific table

```
CREATE TABLE acl (  
    end1 integer, end2 integer, allow integer  
);  
CREATE TABLE server (uid integer);
```

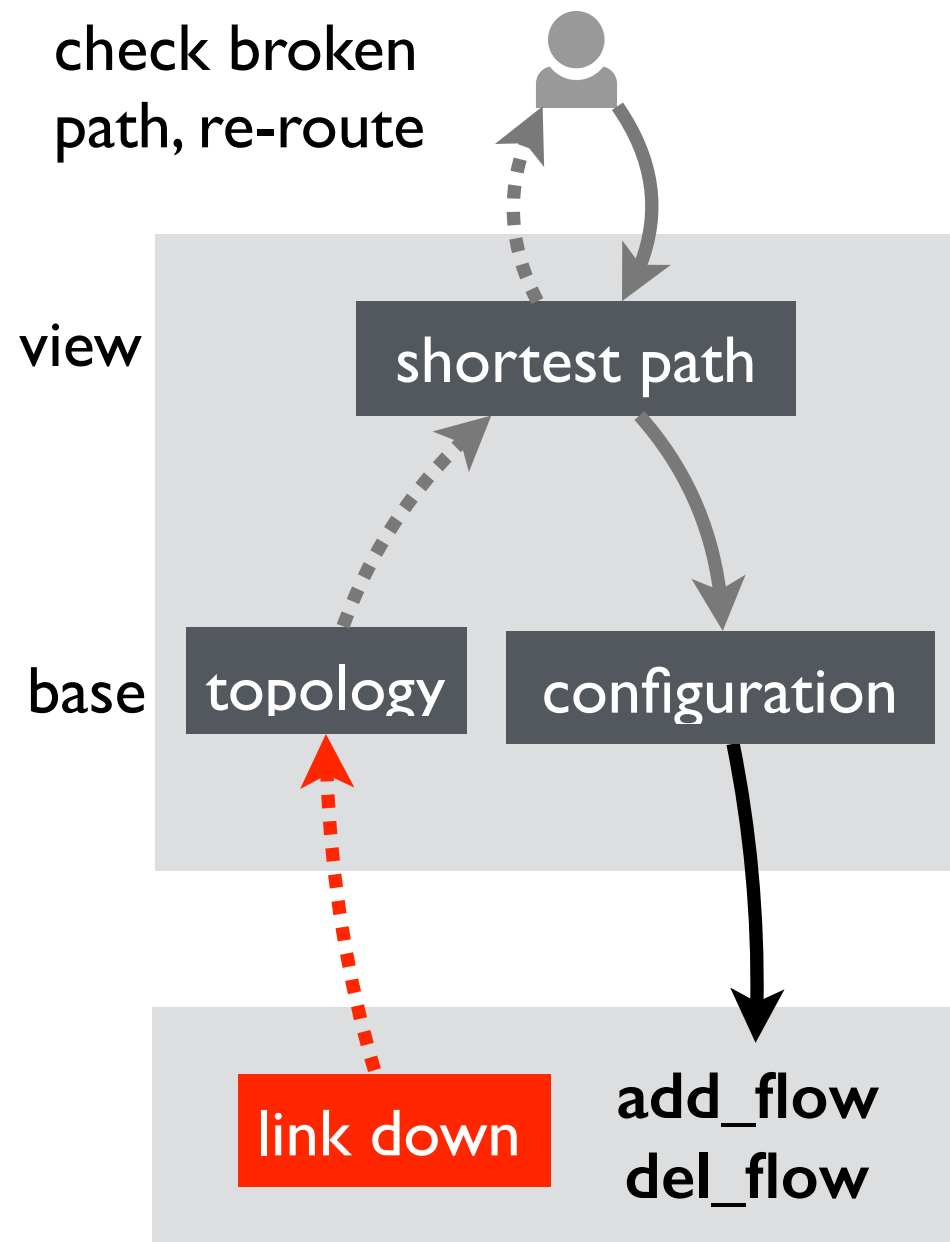
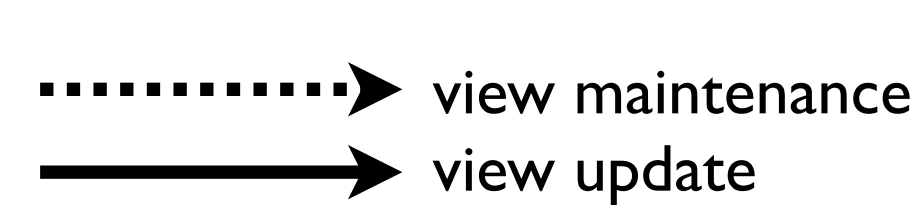
control loop: monitoring firewall view and repairing violation

```
CREATE VIEW acl_violation AS (  
    SELECT fid  
    FROM tm  
    WHERE FW = 1 AND  
        (src, dst) NOT IN  
        (SELECT end1, end2 FROM acl)  
);
```

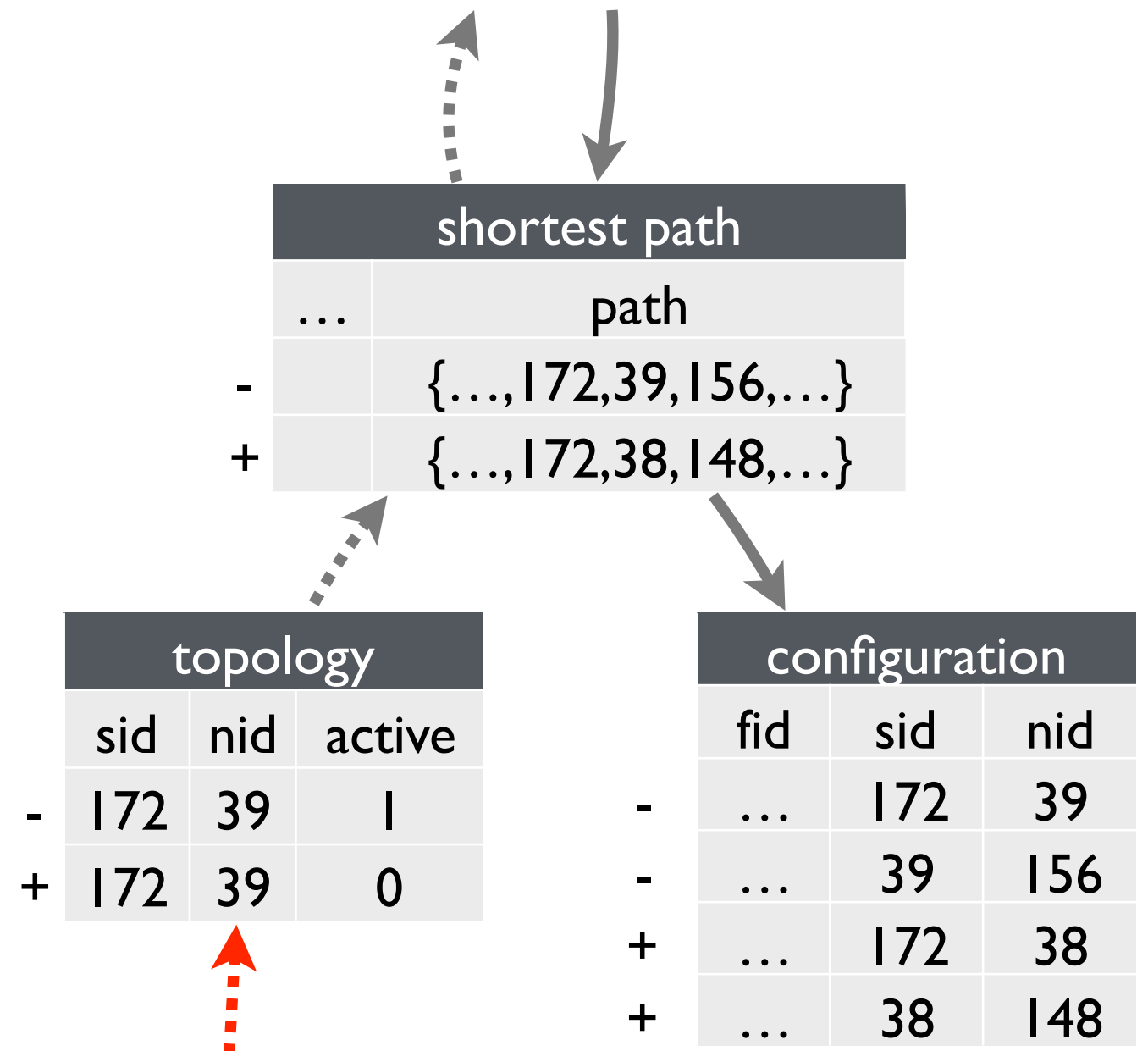
```
CREATE RULE acl_repair AS  
    ON DELETE TO acl_violation  
    DO INSTEAD  
        DELETE FROM tm WHERE fid = OLD.fid;
```

many more: routing, stateful firewall, service chain policy
between endpoint groups ...

orchestration across representations



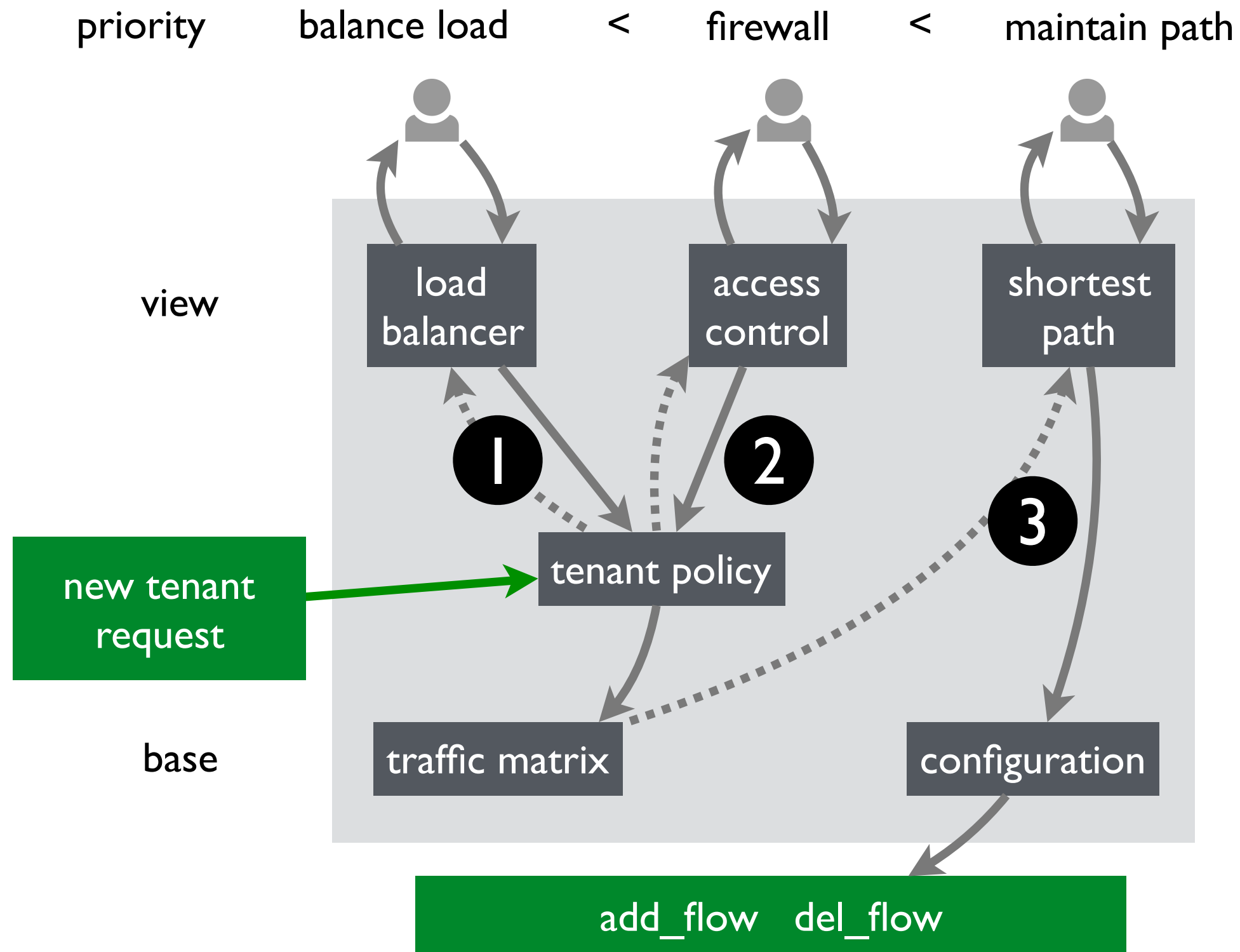
shortest path rule:
upon broken path, re-route



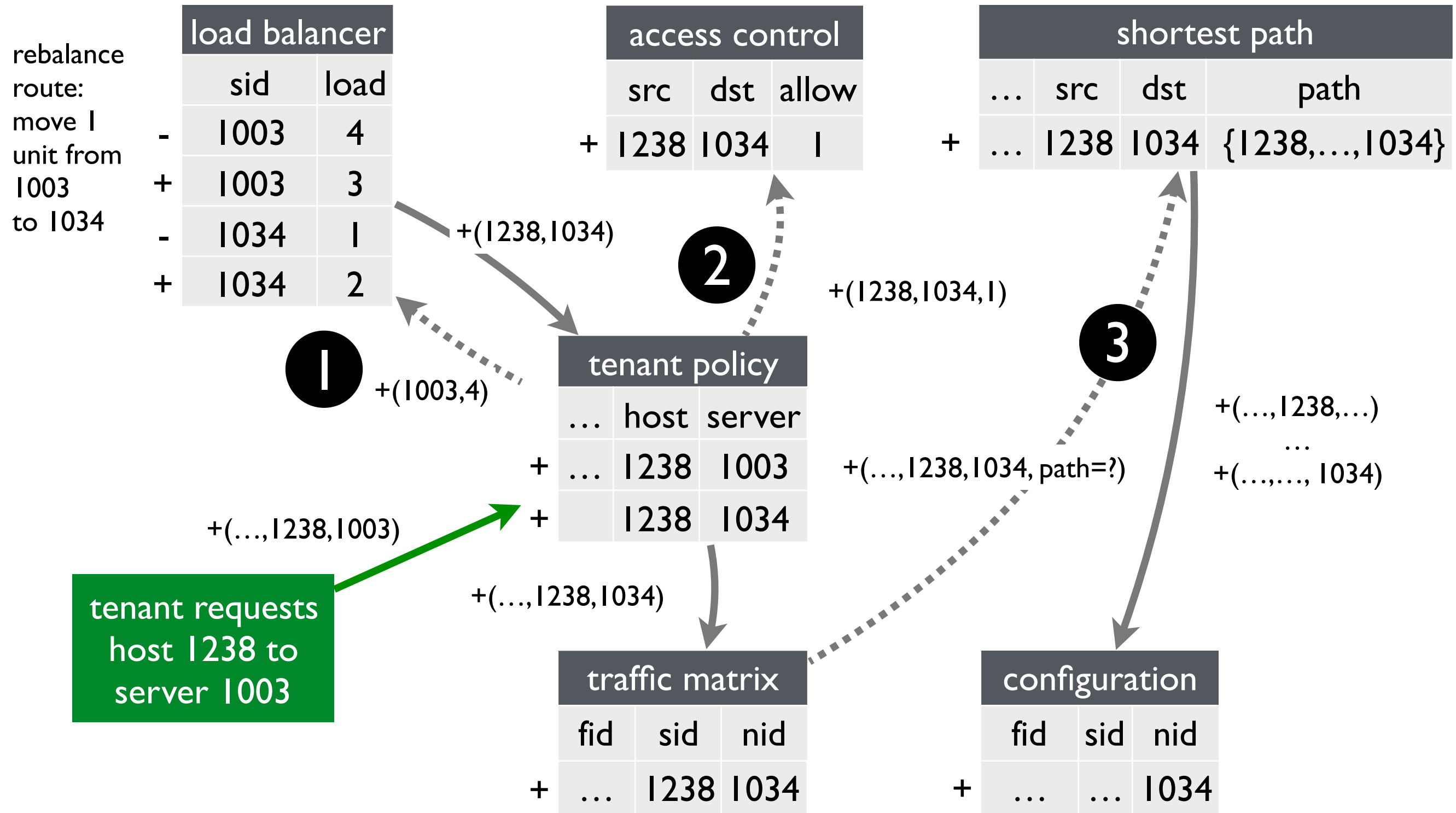
Mininet link (172,39) down

orchestrated updates: re route via (172, 38)

orchestration across applications



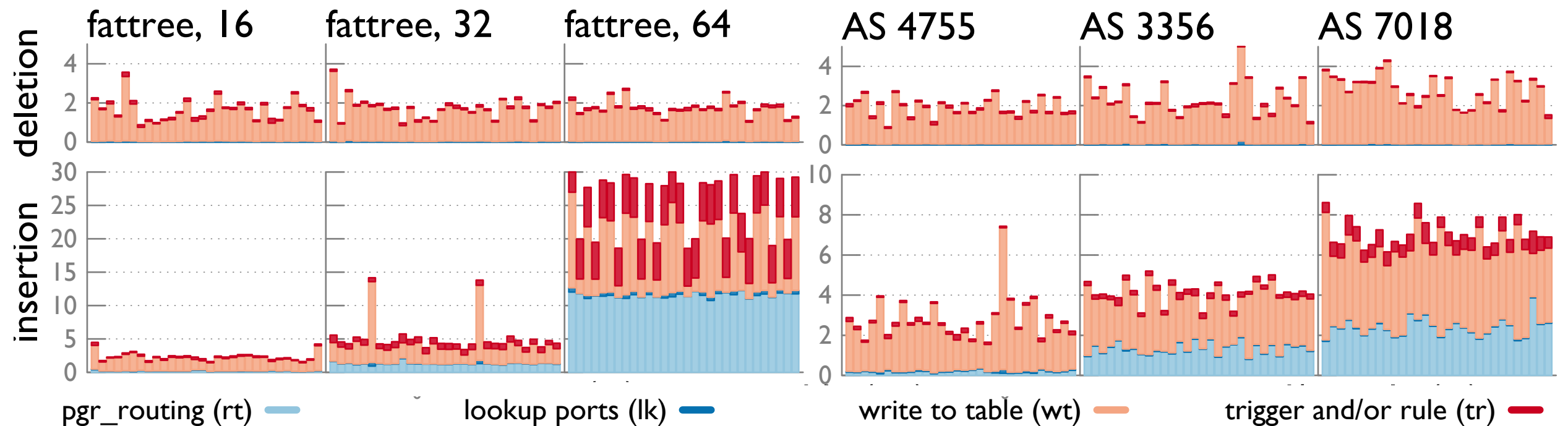
orchestration across applications



orchestrated updates: install alternative route that is load-balanced and safe

evaluation

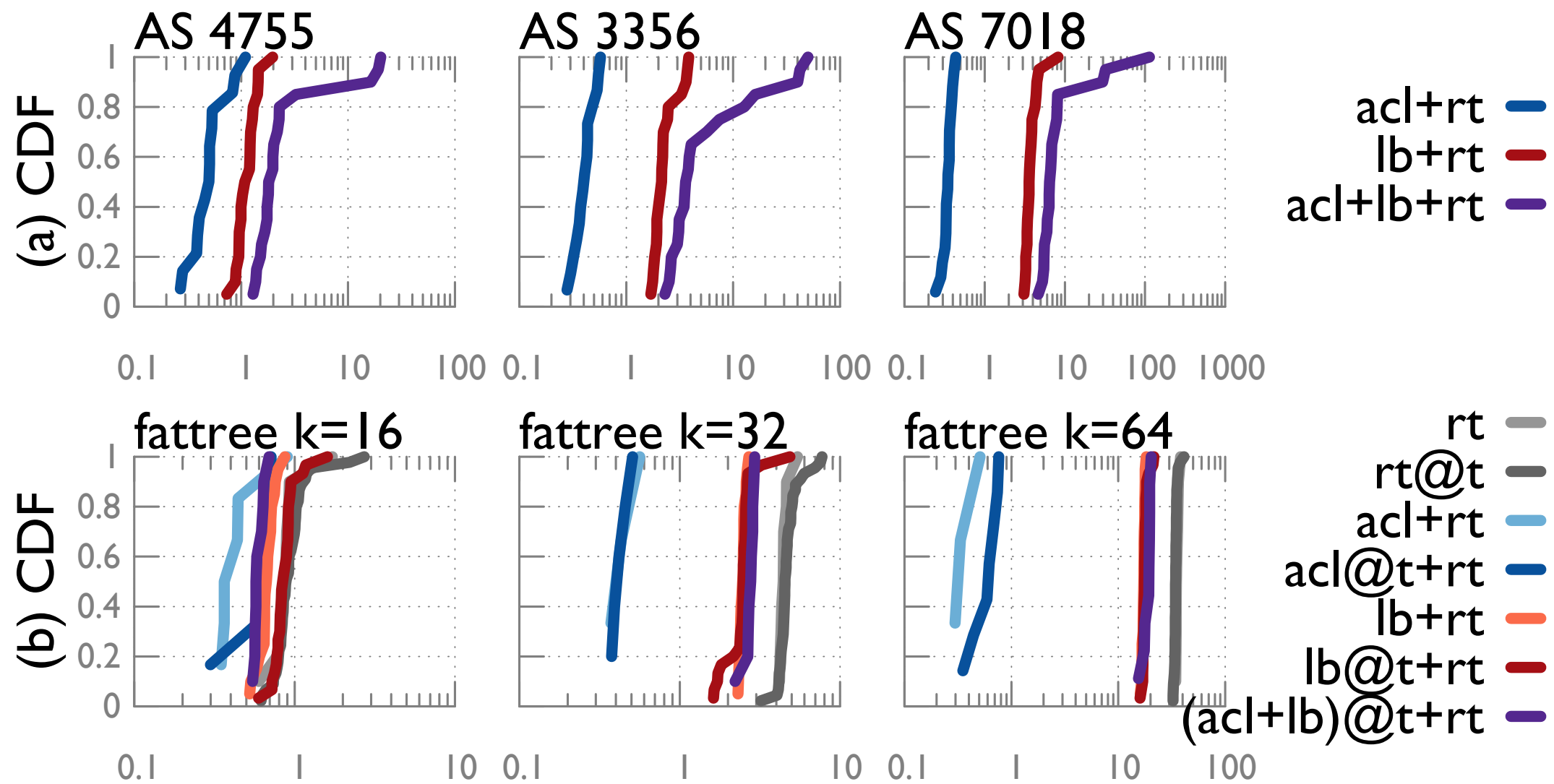
profiling database delay — route insertion/deletion



fat-tree			ISP		
k	switches	links	AS#	nodes	links
16	320	3072	4755	142	258
32	1280	24576	3356	1772	13640
64	5120	196608	7018	25382	11292
			2914	5939	16520

evaluation

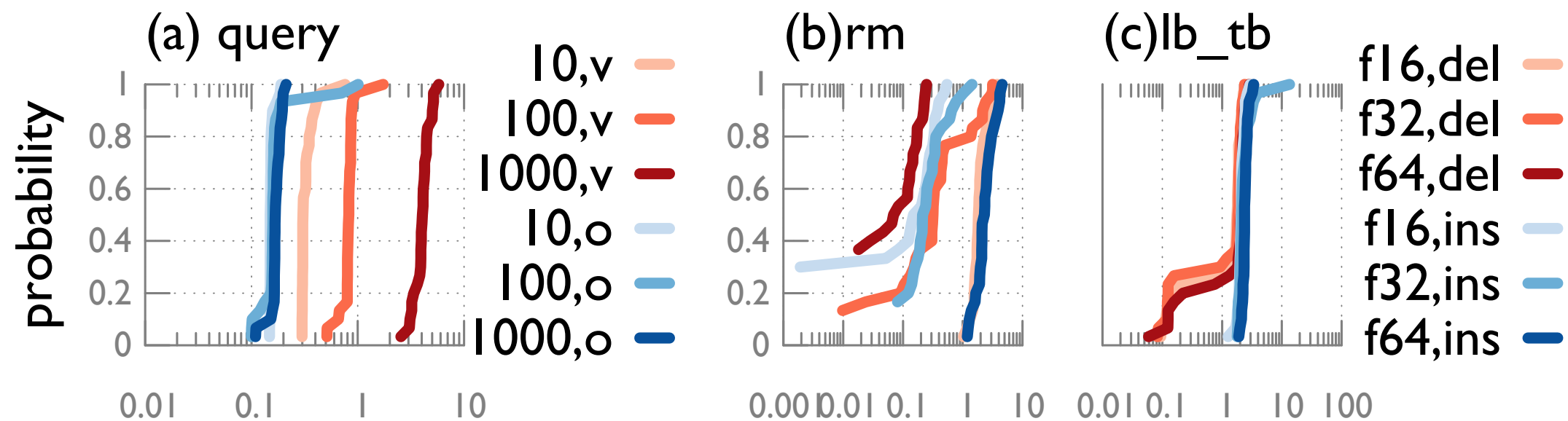
orchestrating access control(acl), load balancer(lb), and routing(rt): normalized per-rule delay (ms)



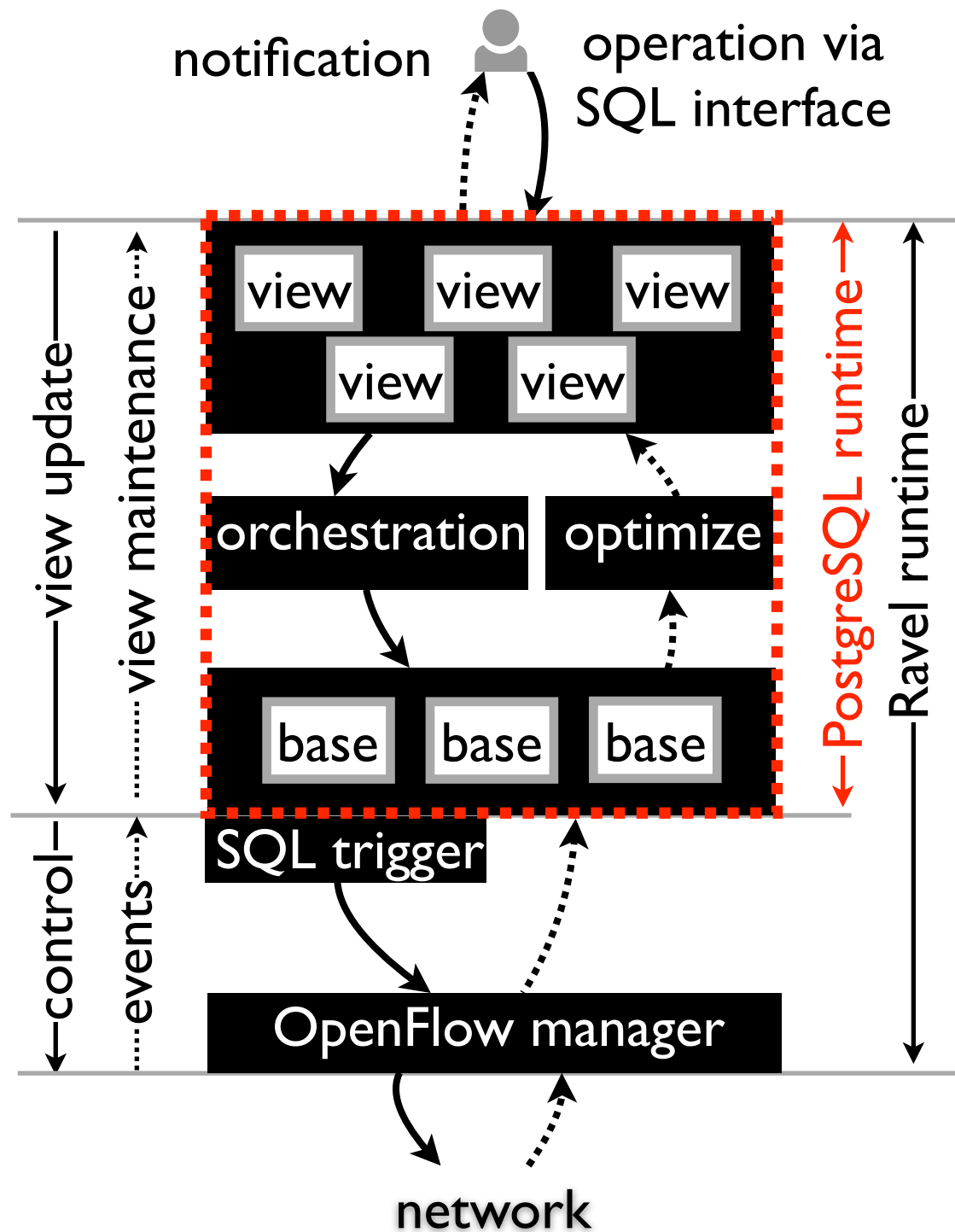
evaluation

optimizing application—materializing views

- faster access to materialized view (a)
- small maintenance delay (b,c)



conclusion



attractive features

- ad-hoc programmable abstraction via views
- orchestration across abstractions via view mechanism
- orchestration across applications via data mediation
- network control via SQL

promising performance
even on large networks

looking forward

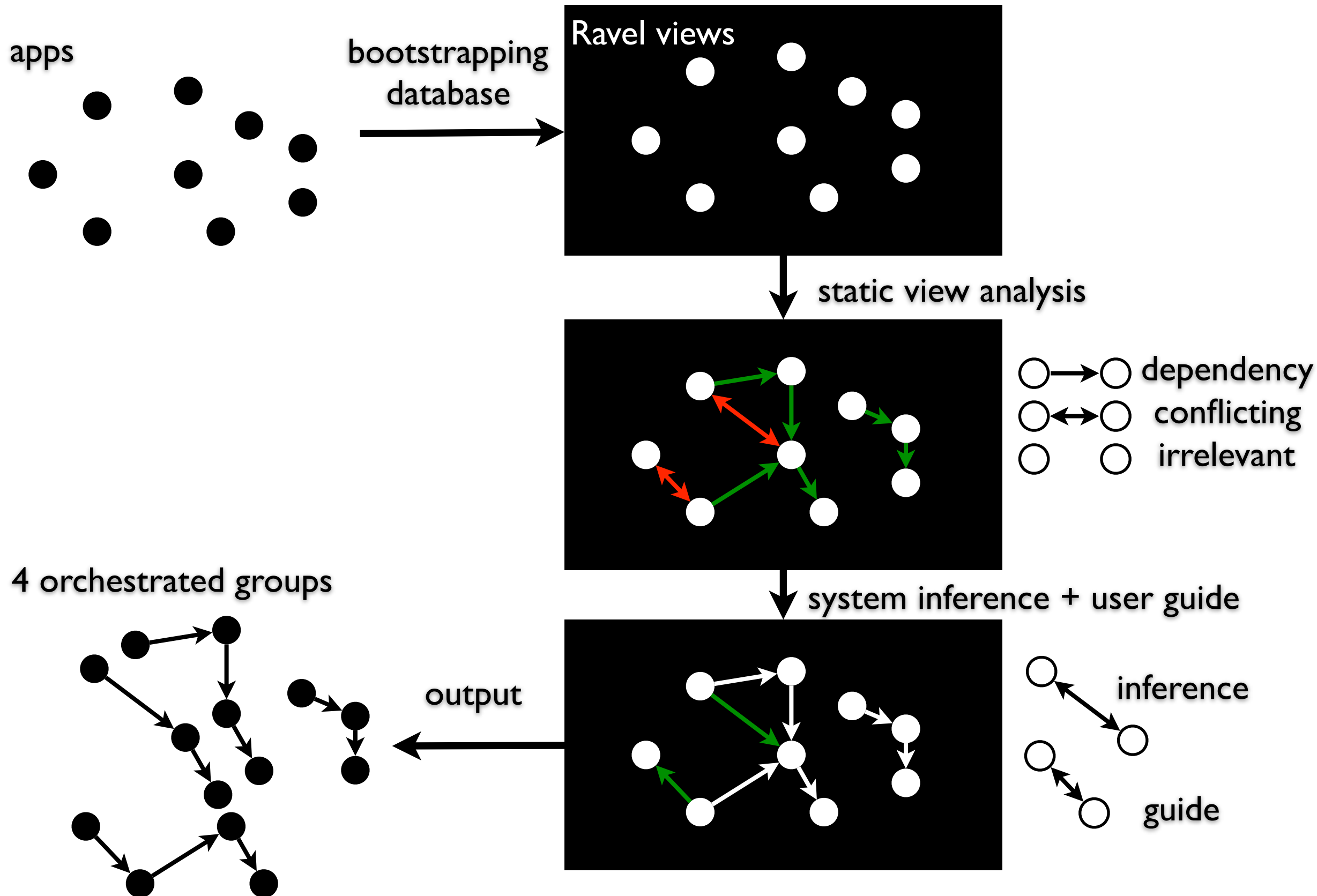
use of standard SQL database enables direct application of many database theories and facilities

- revisit concurrency and recovery control
 - transaction processing
- revisit state distribution
 - distributed and federated database

ongoing work

- synthesizing orchestration

synthesizing orchestration



thanks



playtime

website (quick start, tutorials ...)

<https://ravel-net.org>

github

<https://github.com/ravel-net>

download *Ravel* (vm image)

<https://cis.temple.edu/~adw/ravel/ravelvm.zip>