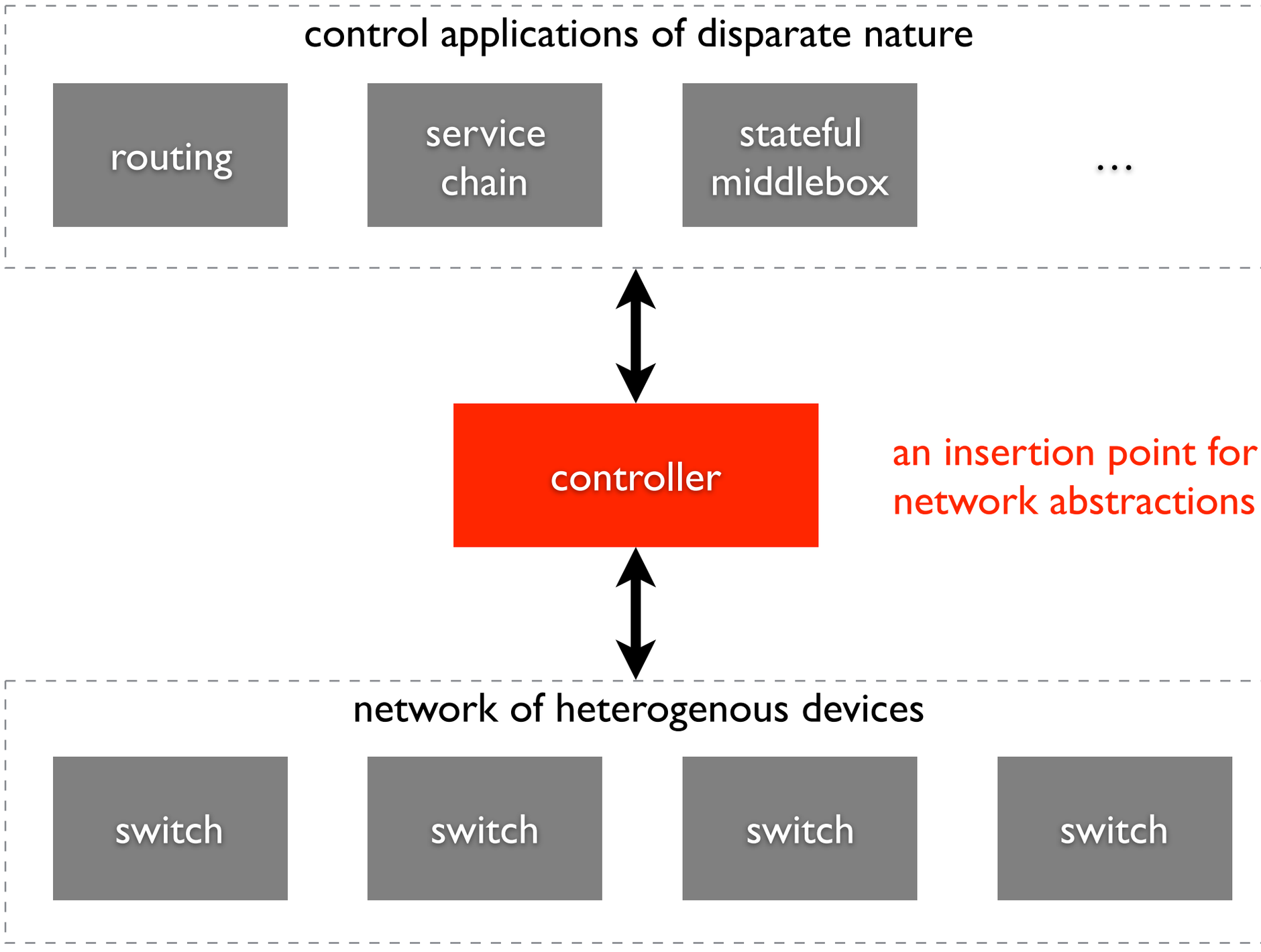
Ravel: a database-defined network

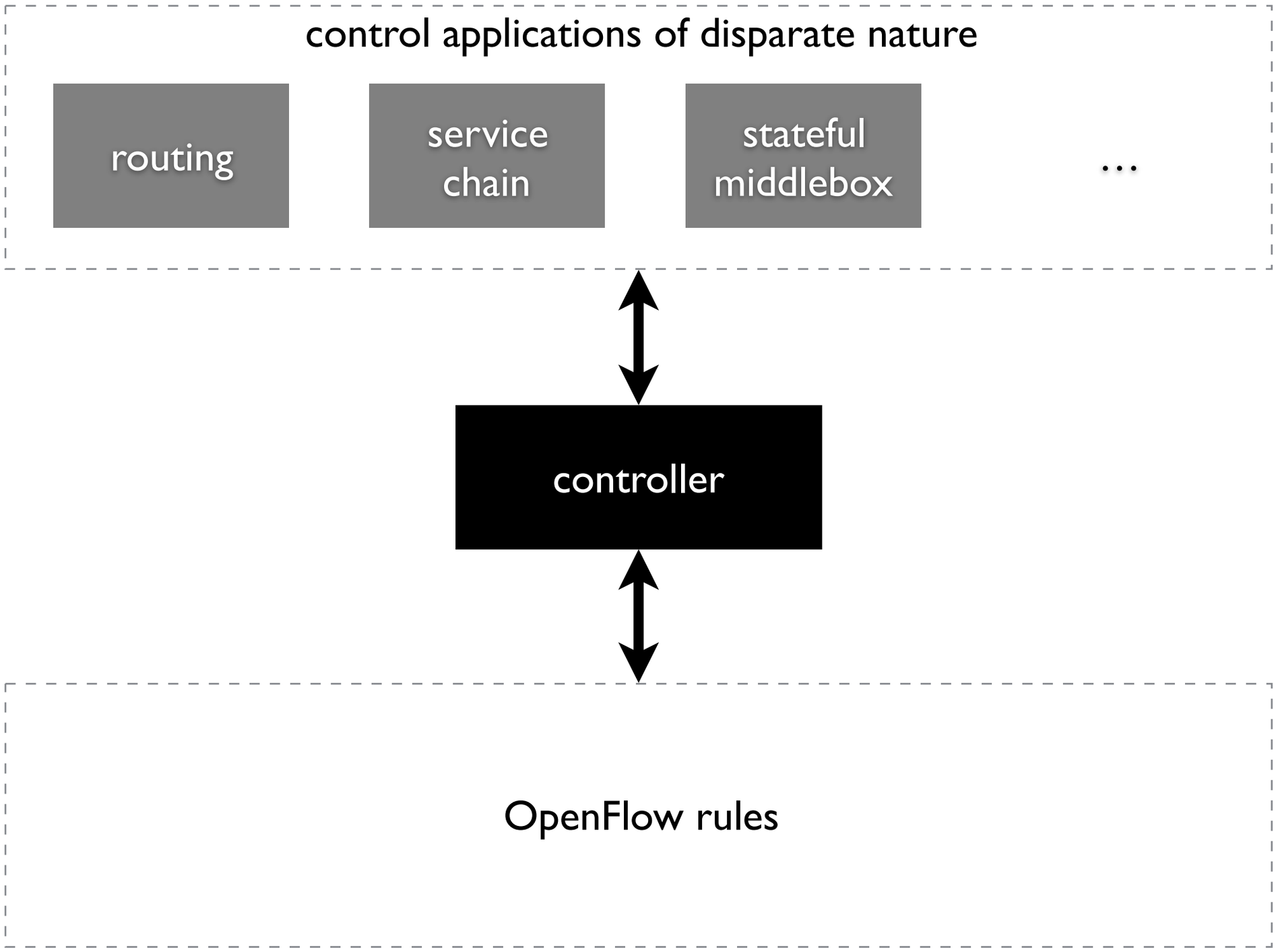
Anduo Wang Xueyuan Mei Jason Croft
Matthew Caesar Brighten Godfrey

March 14, 2016 SOSR'16

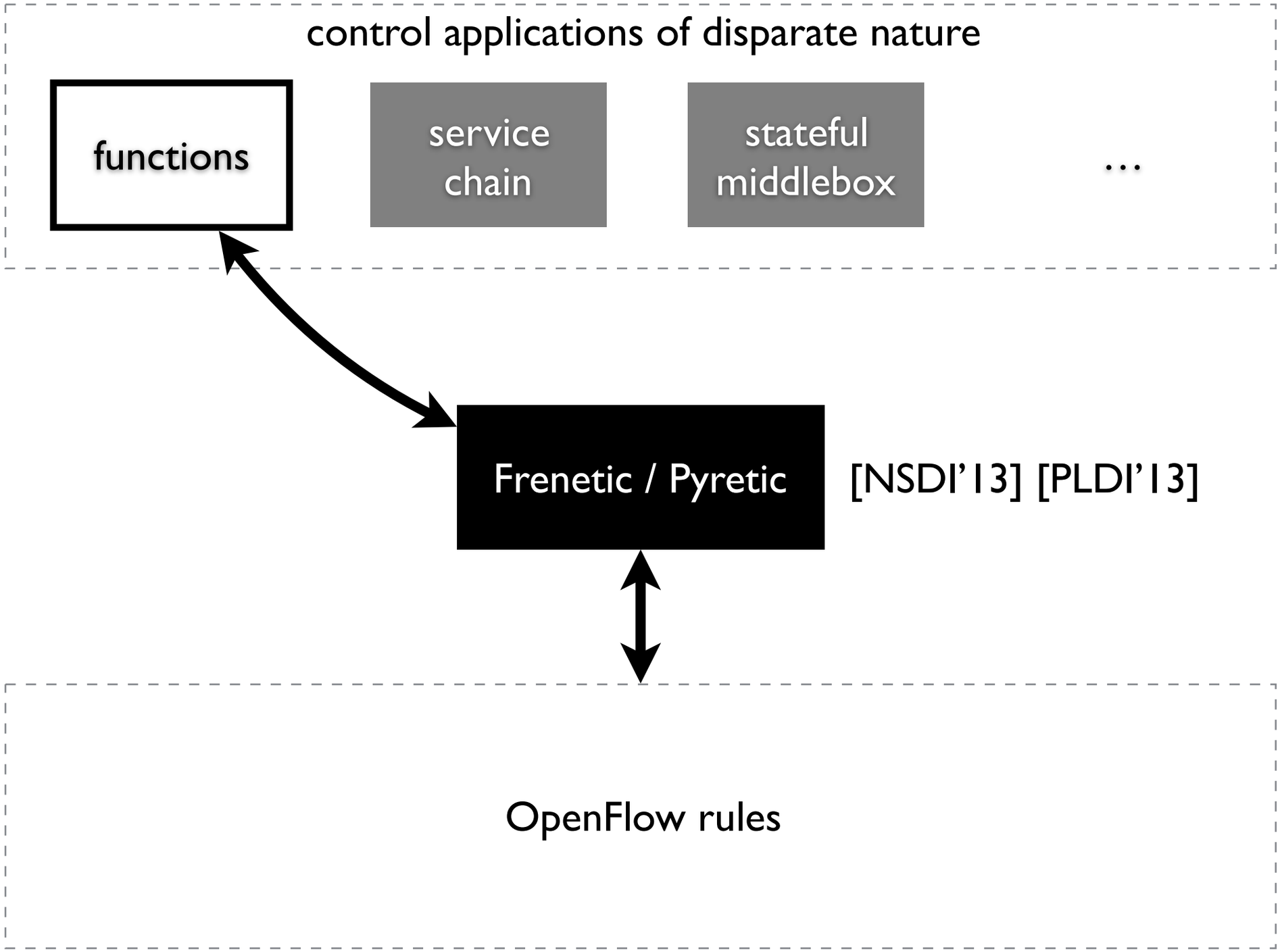
software-defined network



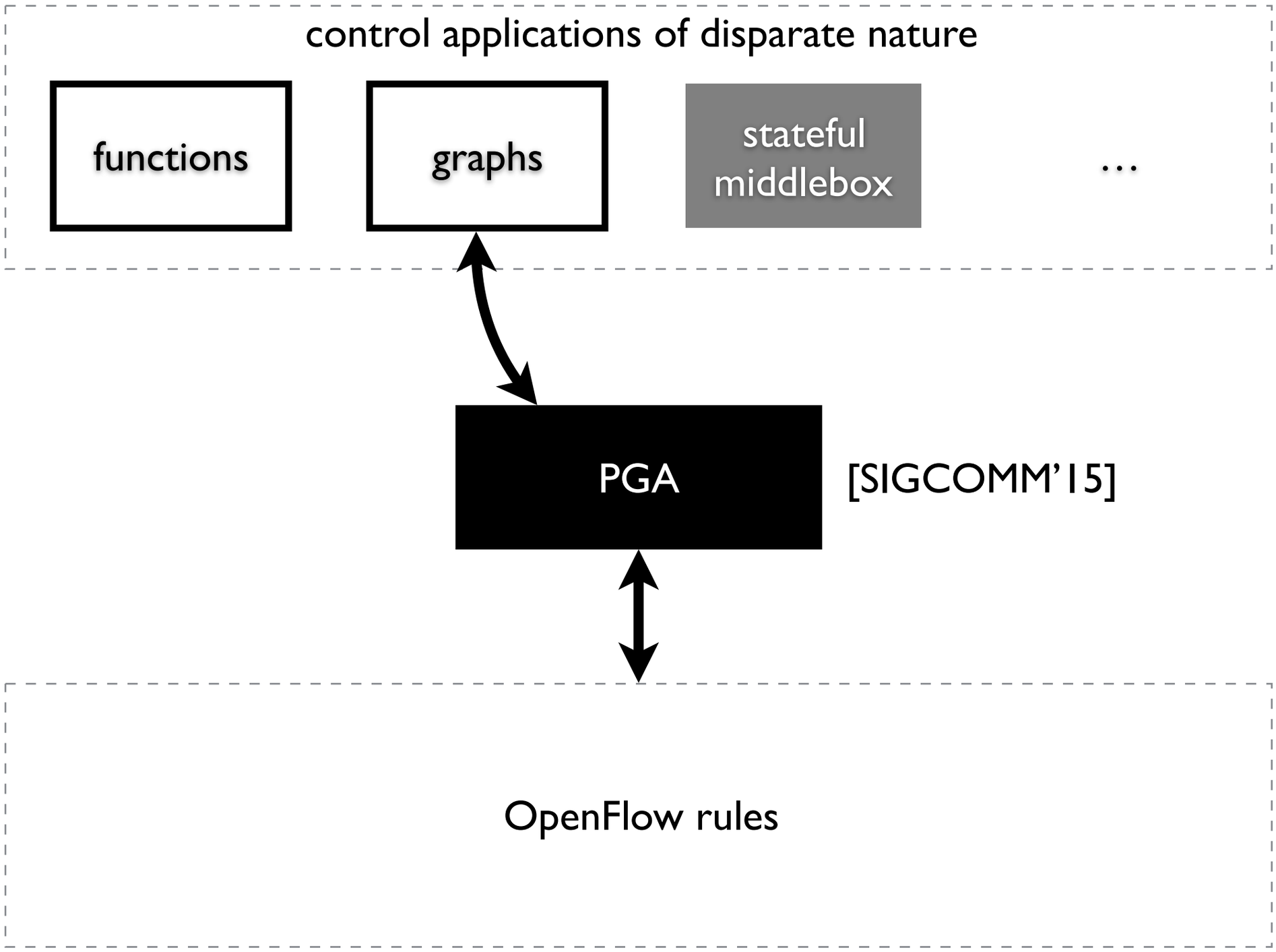
abstractions



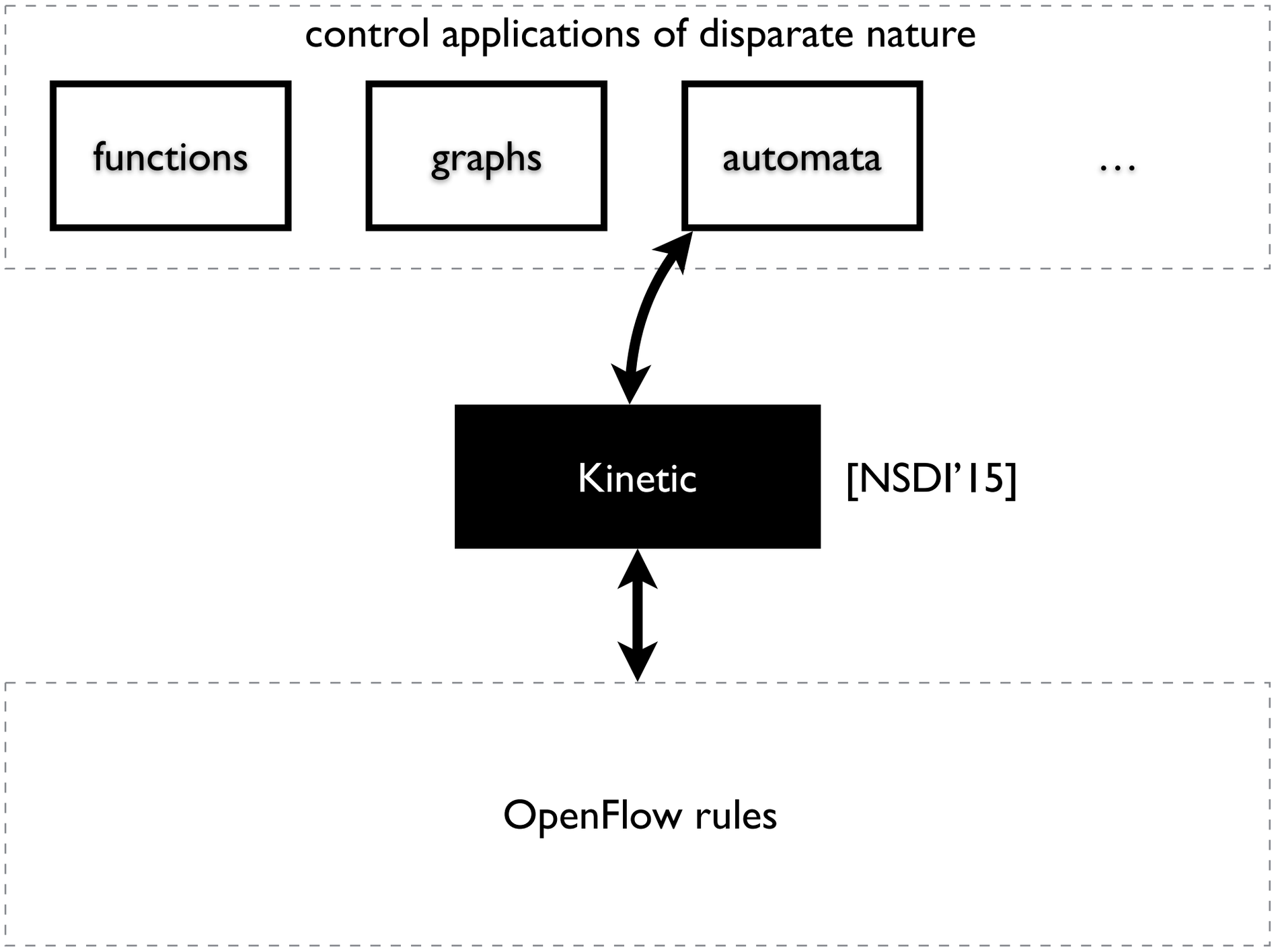
abstractions



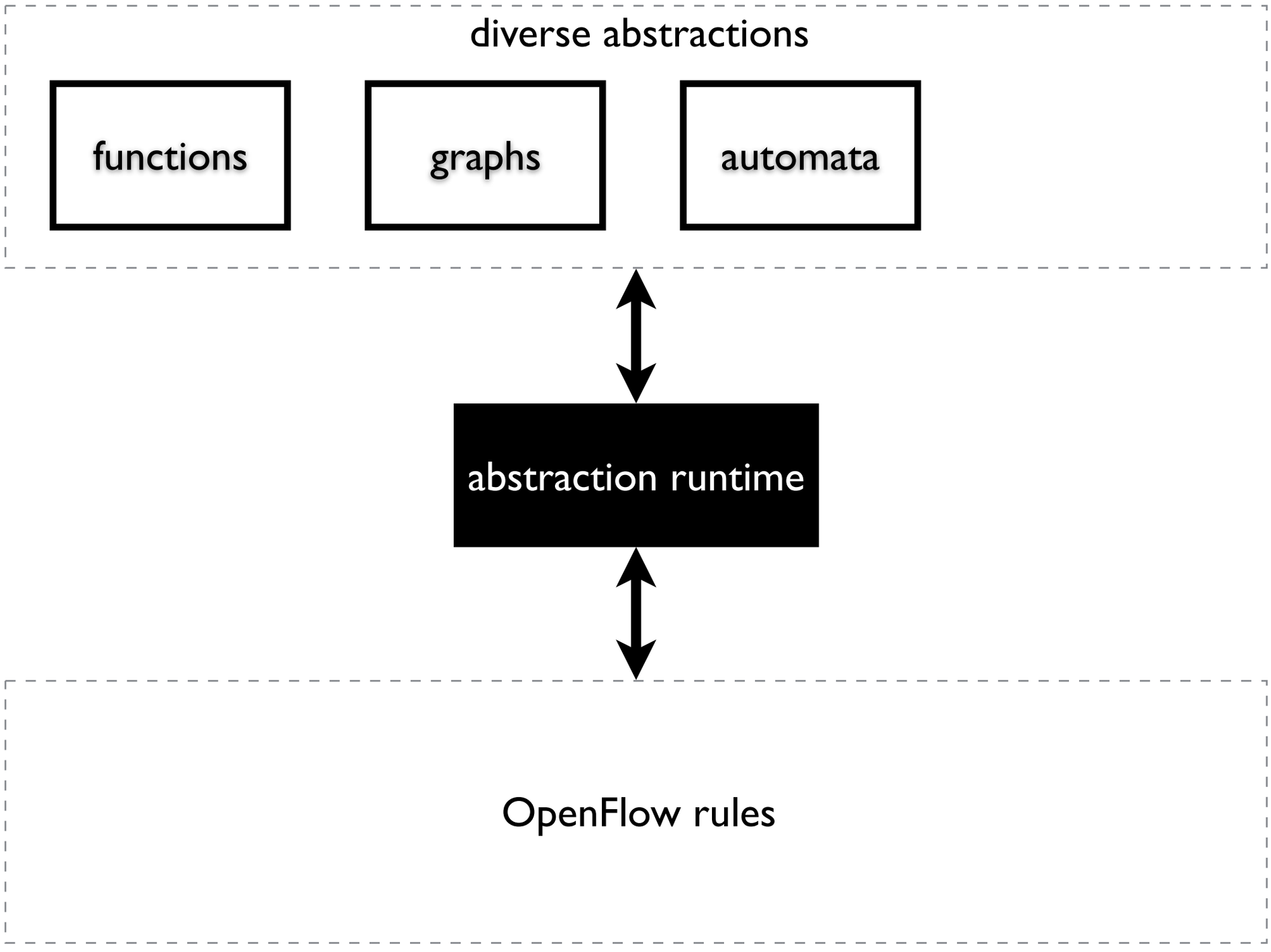
abstractions



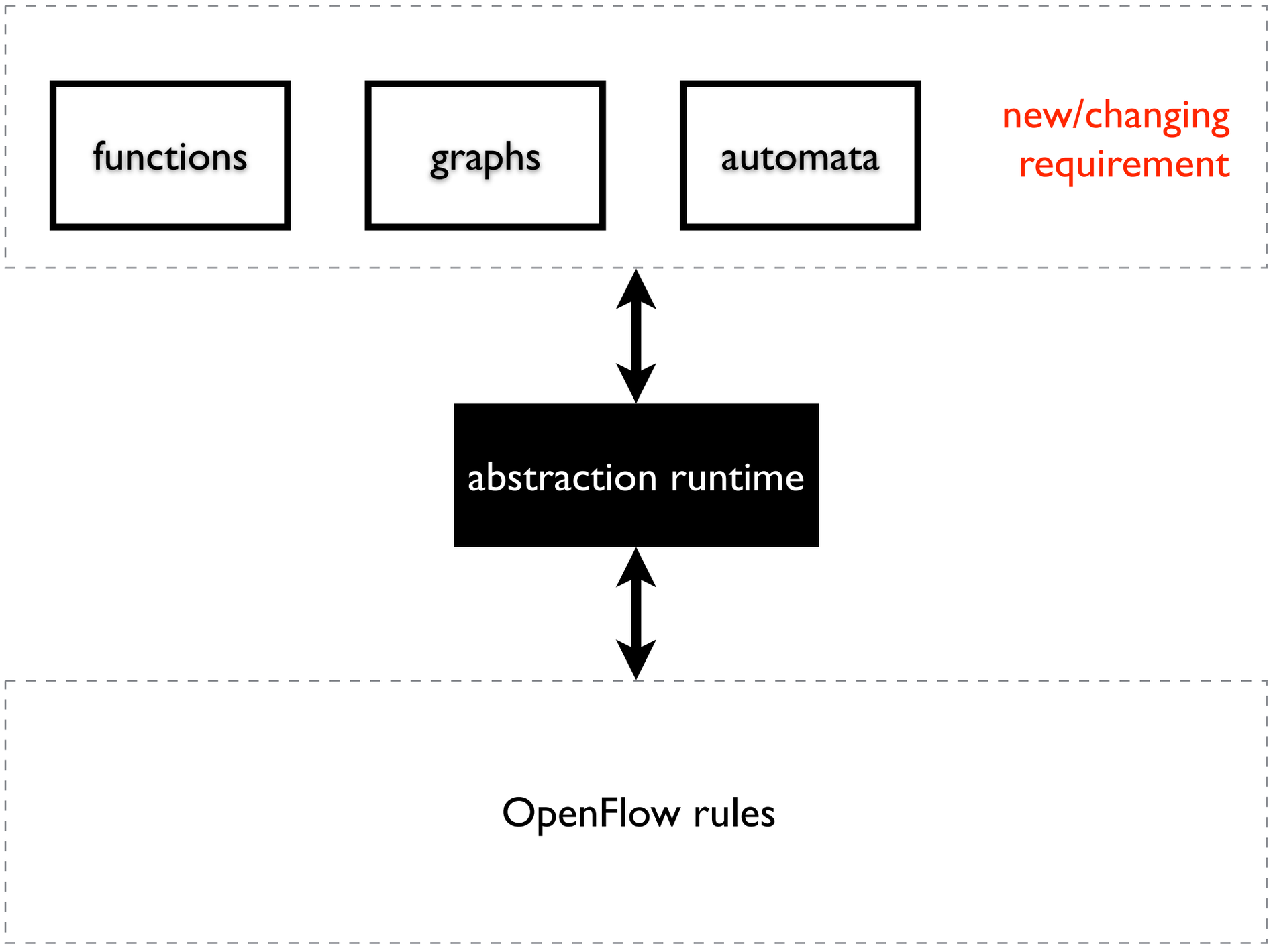
abstractions



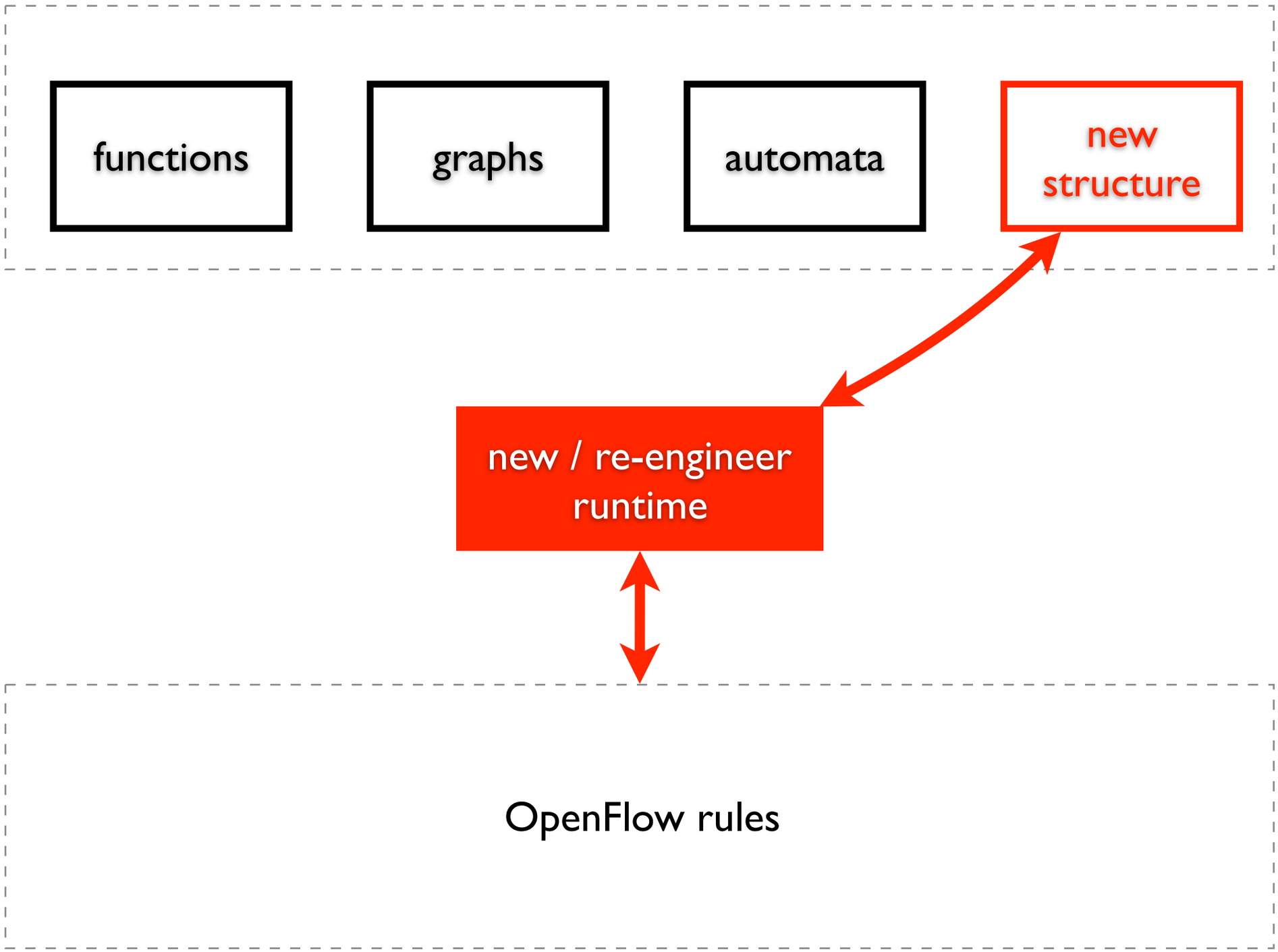
abstractions



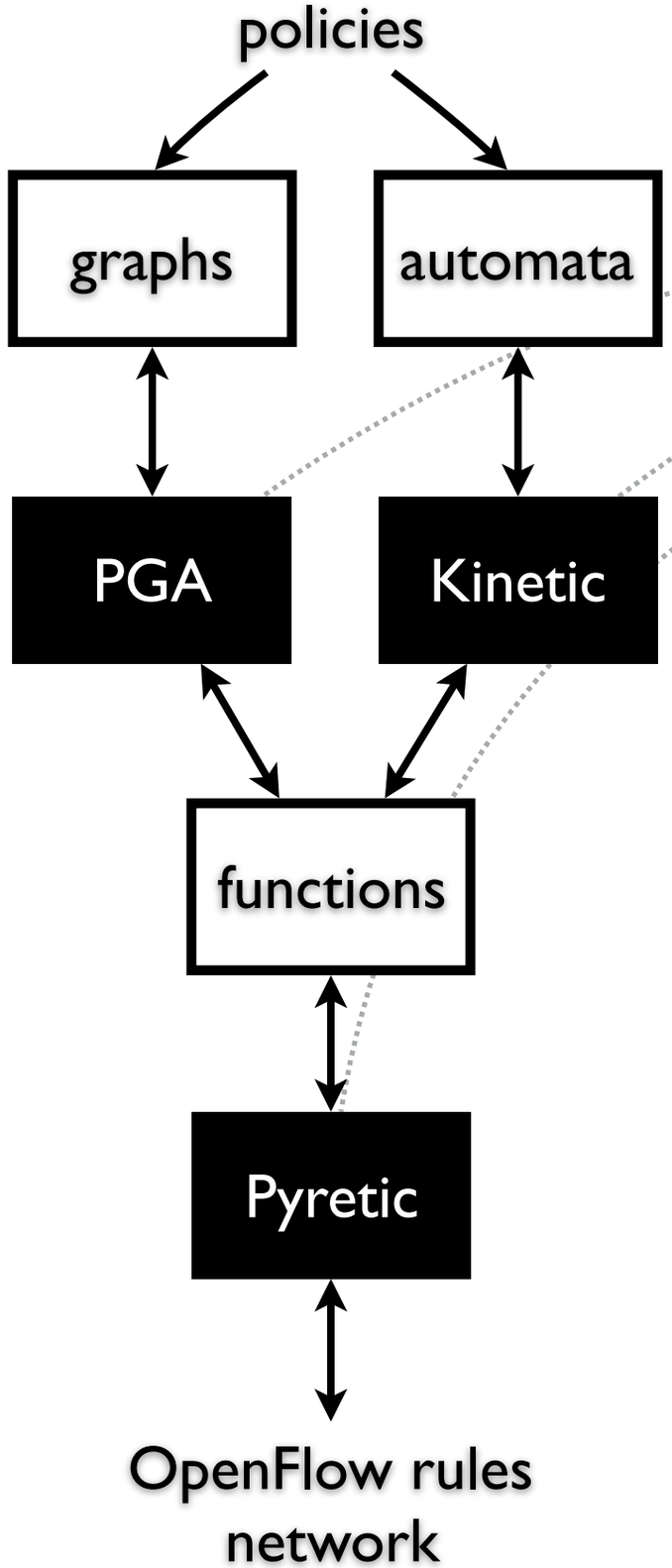
but network keep evolving



but network keep evolving

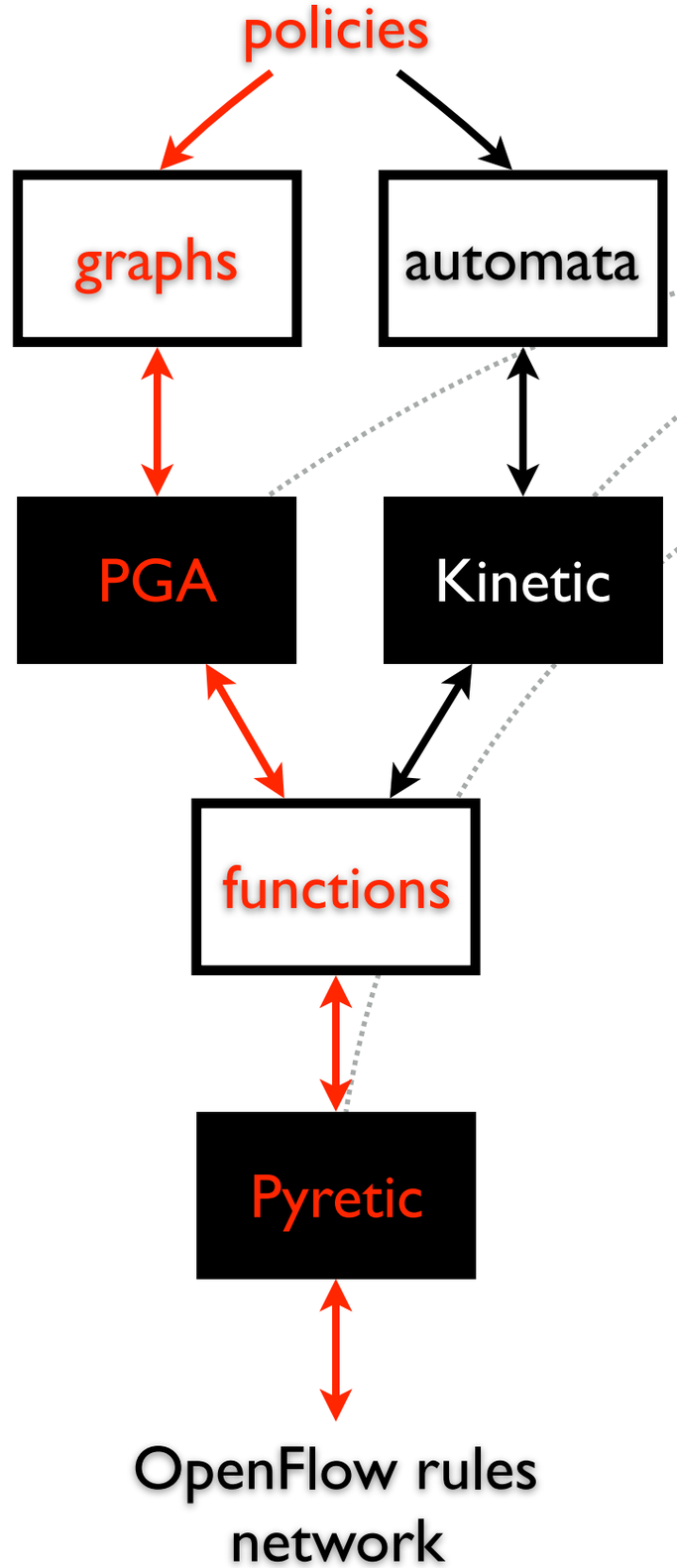


and applications (components) interact



language-level orchestration restricted to each abstraction

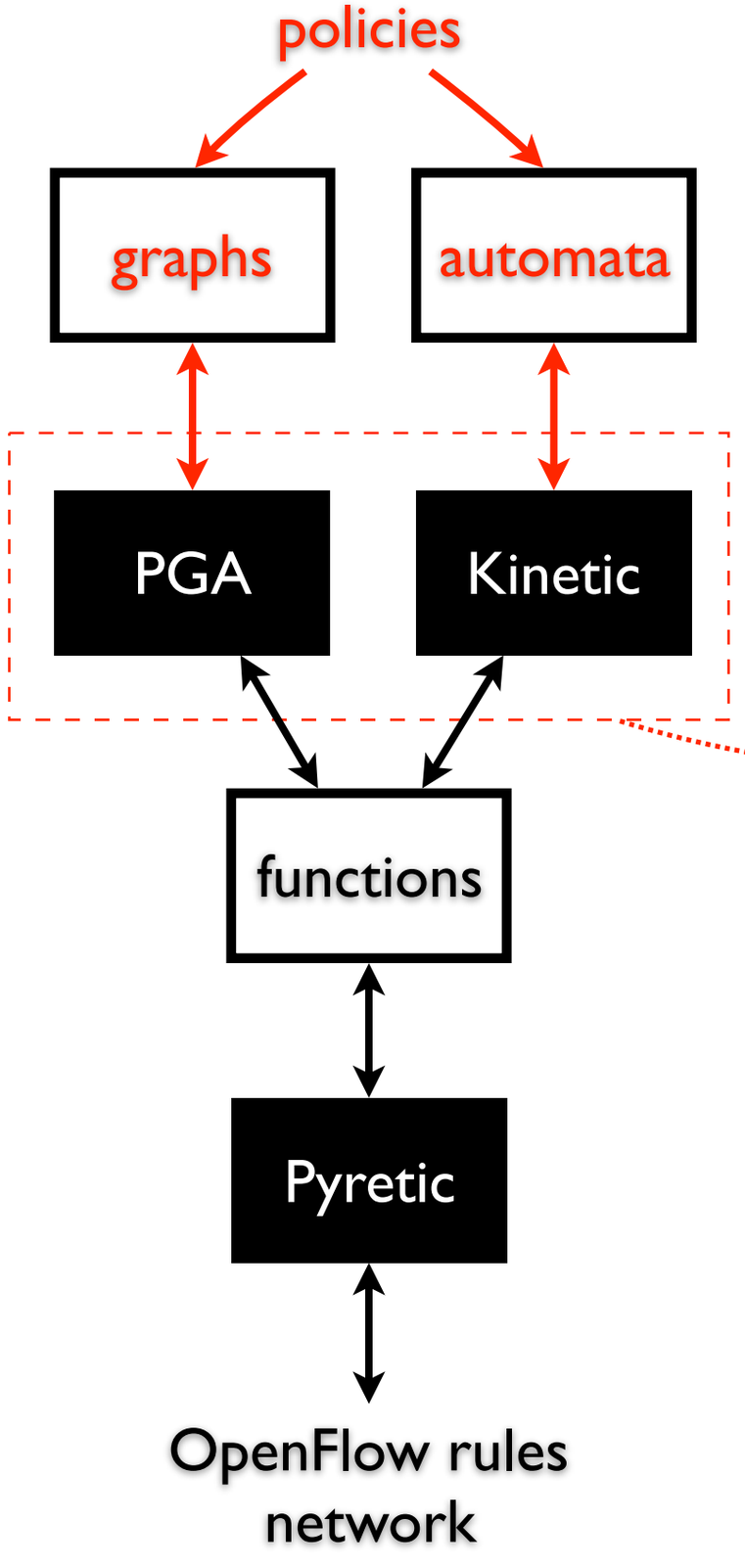
and applications (components) interact



language-level orchestration restricted to each abstraction

composing (+) policy
→ graph $+_{PGA}$ graph
→ function $+_{Pyretic}$ function

and applications (components) interact

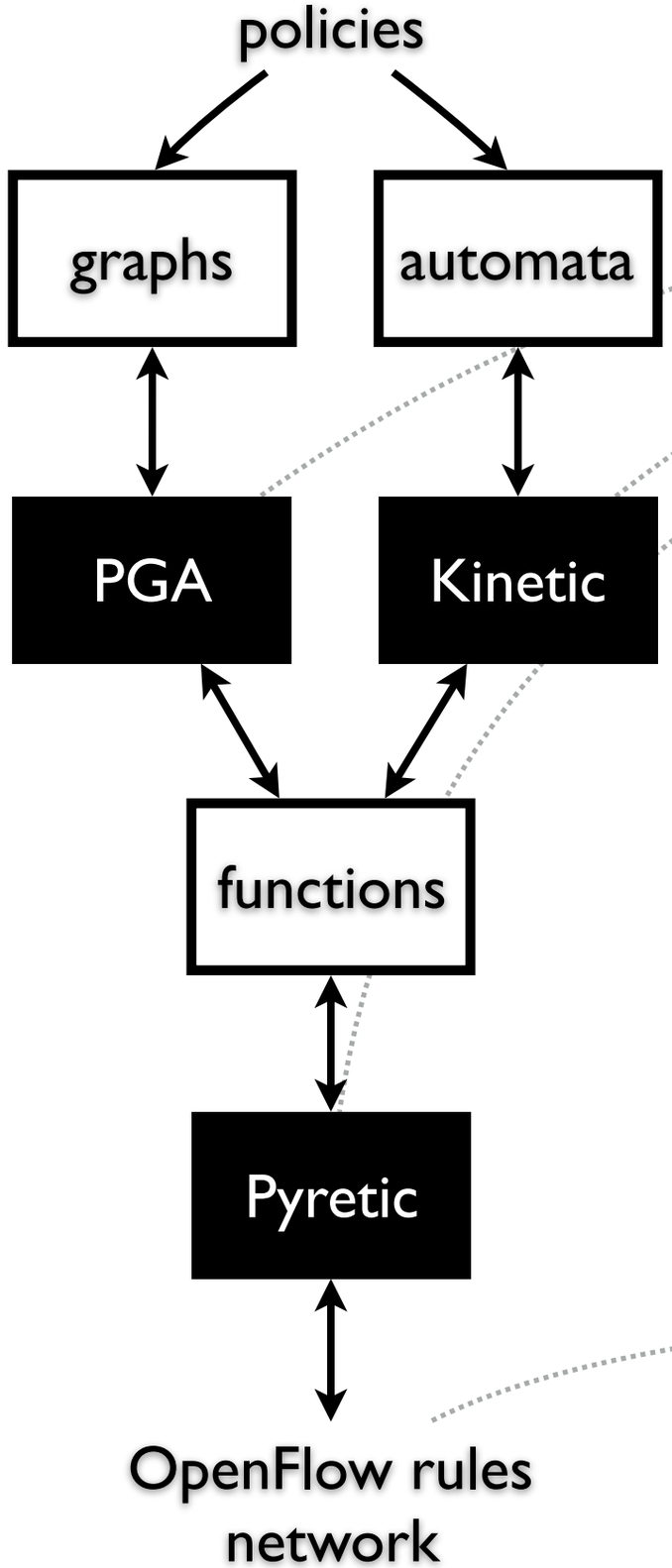


language-level orchestration restricted to each abstraction

composing (+) policy
→ graph +? automata

integrate the runtime by hard-wiring internals

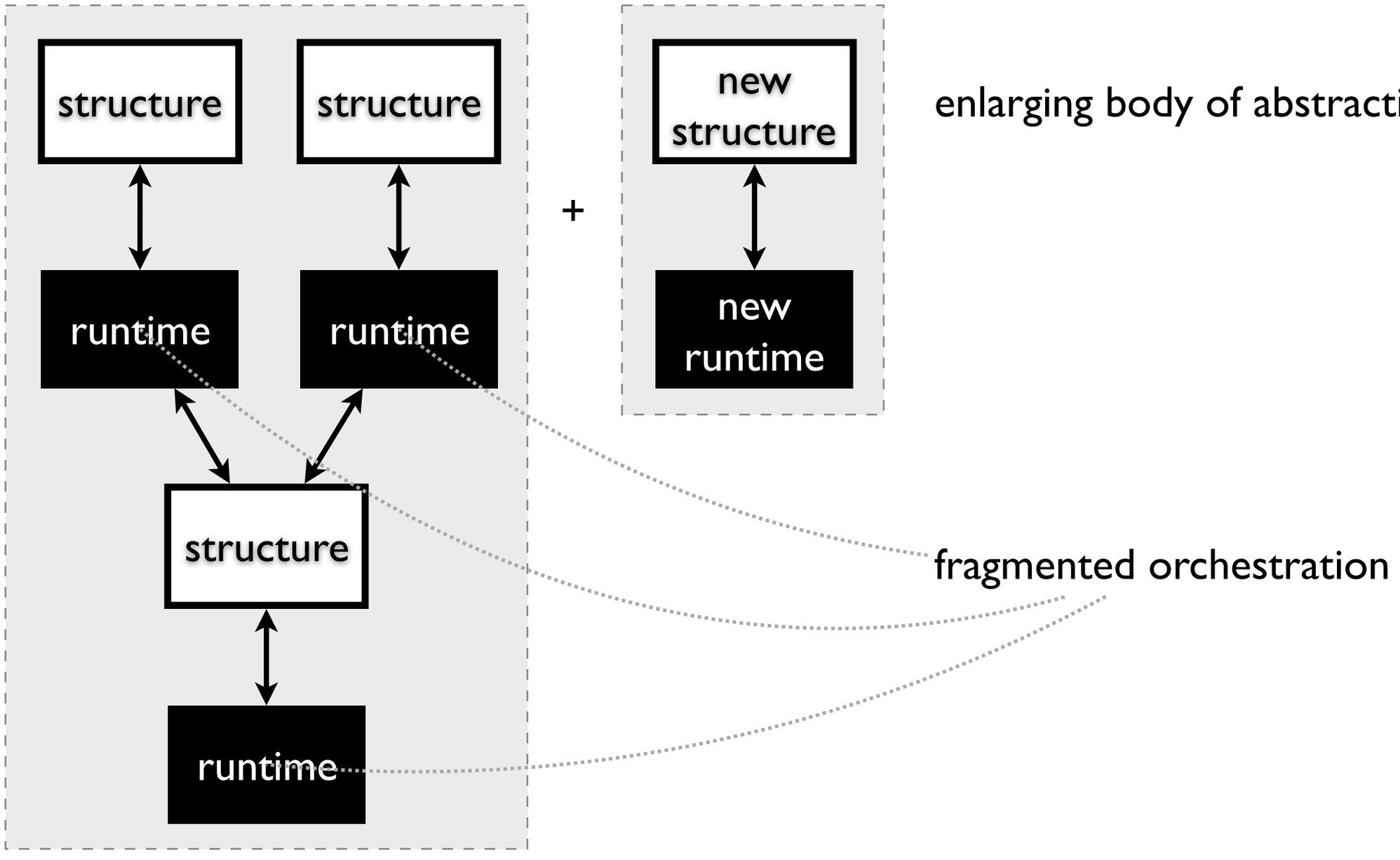
and applications (components) interact



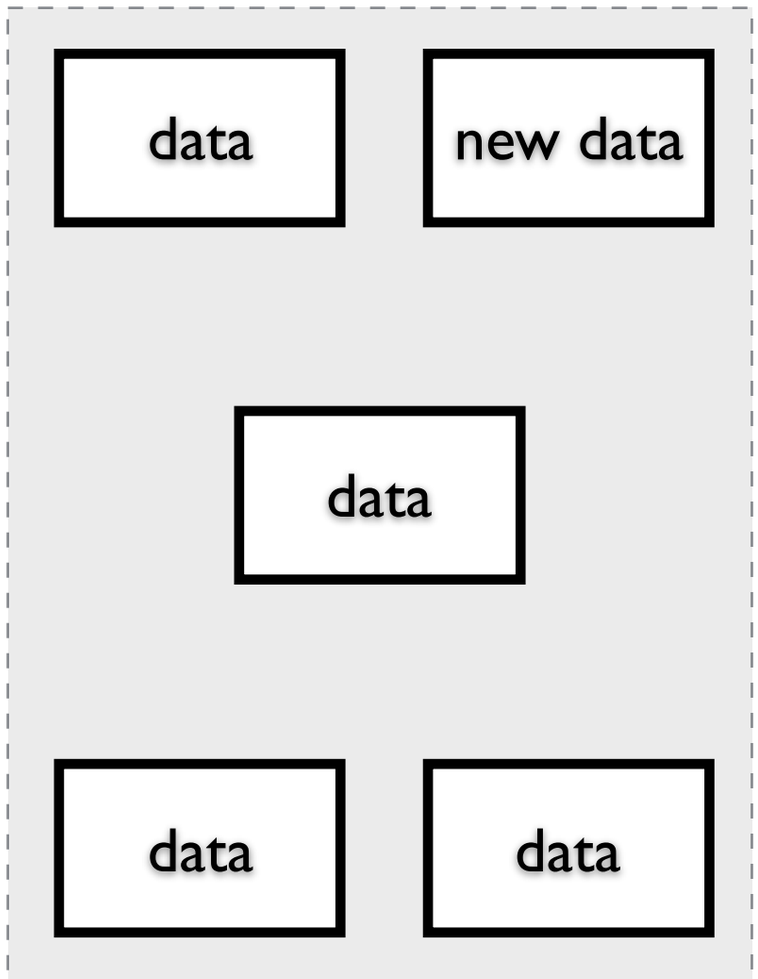
language-level orchestration restricted to each abstraction

abstraction-agonistic coordination often low-level
Co-visor [NSDI'15] statesman [SIGCOMM'14]

current states of abstraction



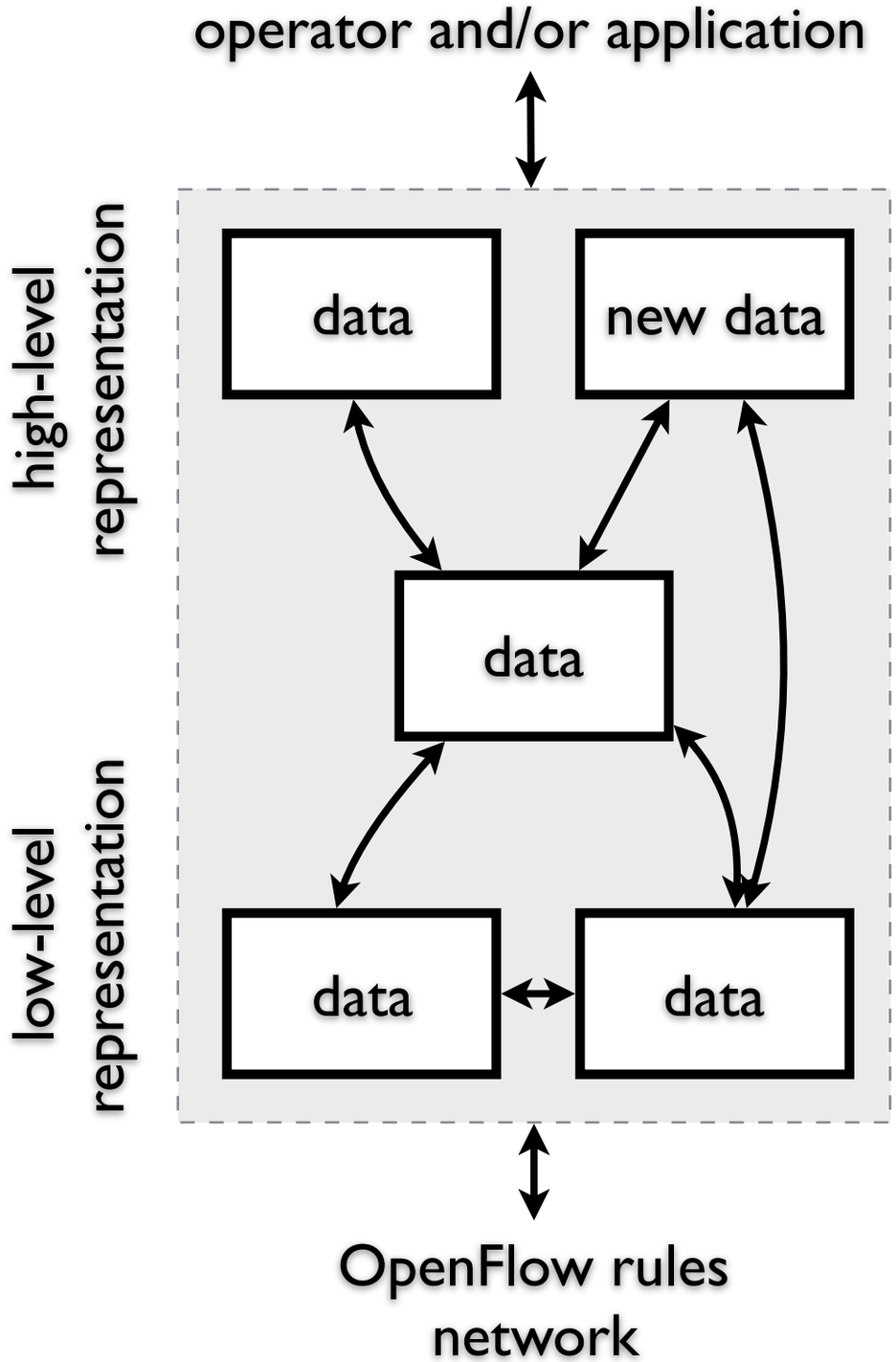
our perspective



SDN control revolves around data representation

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

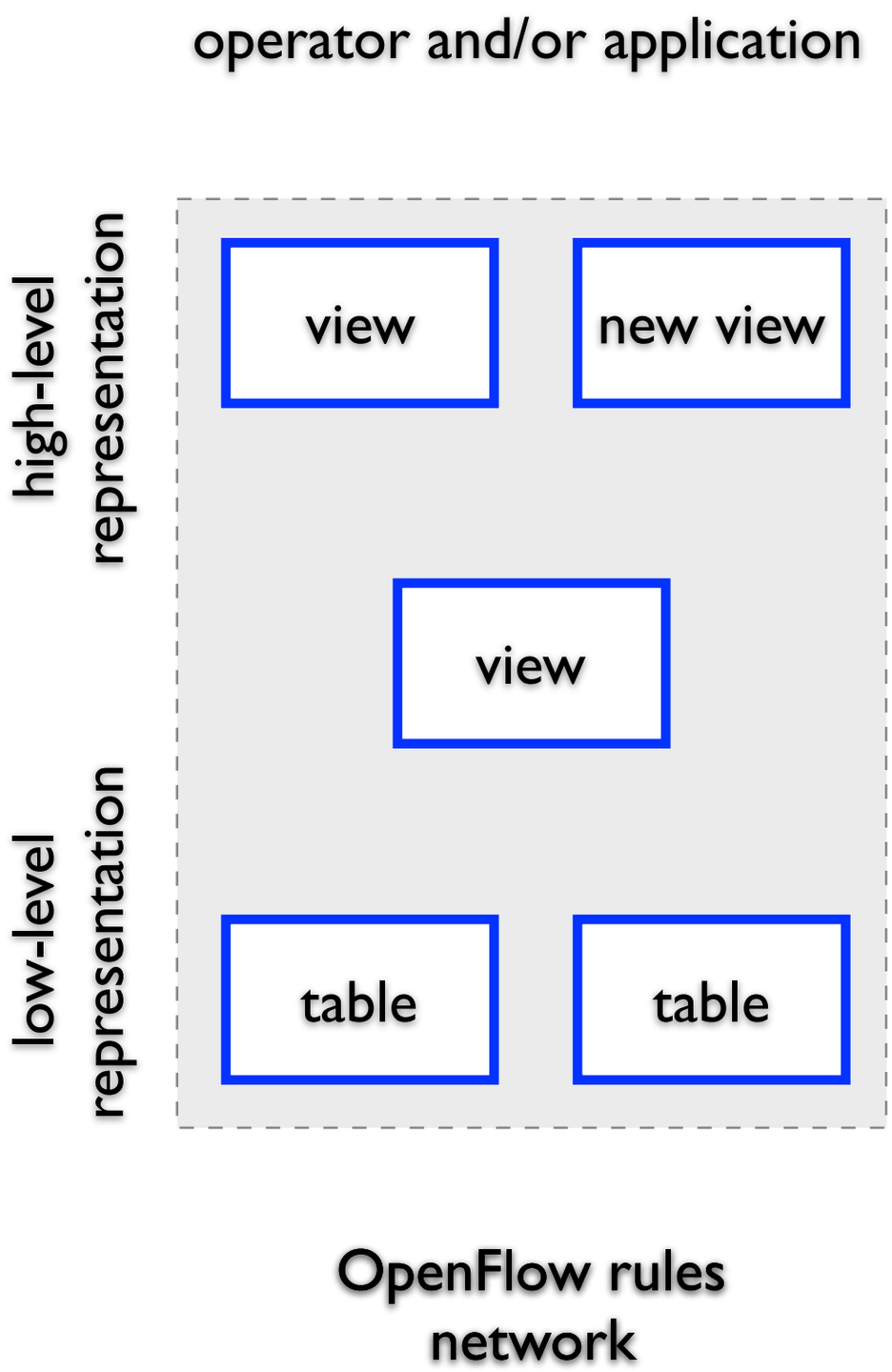
our perspective



SDN control revolves around data representation

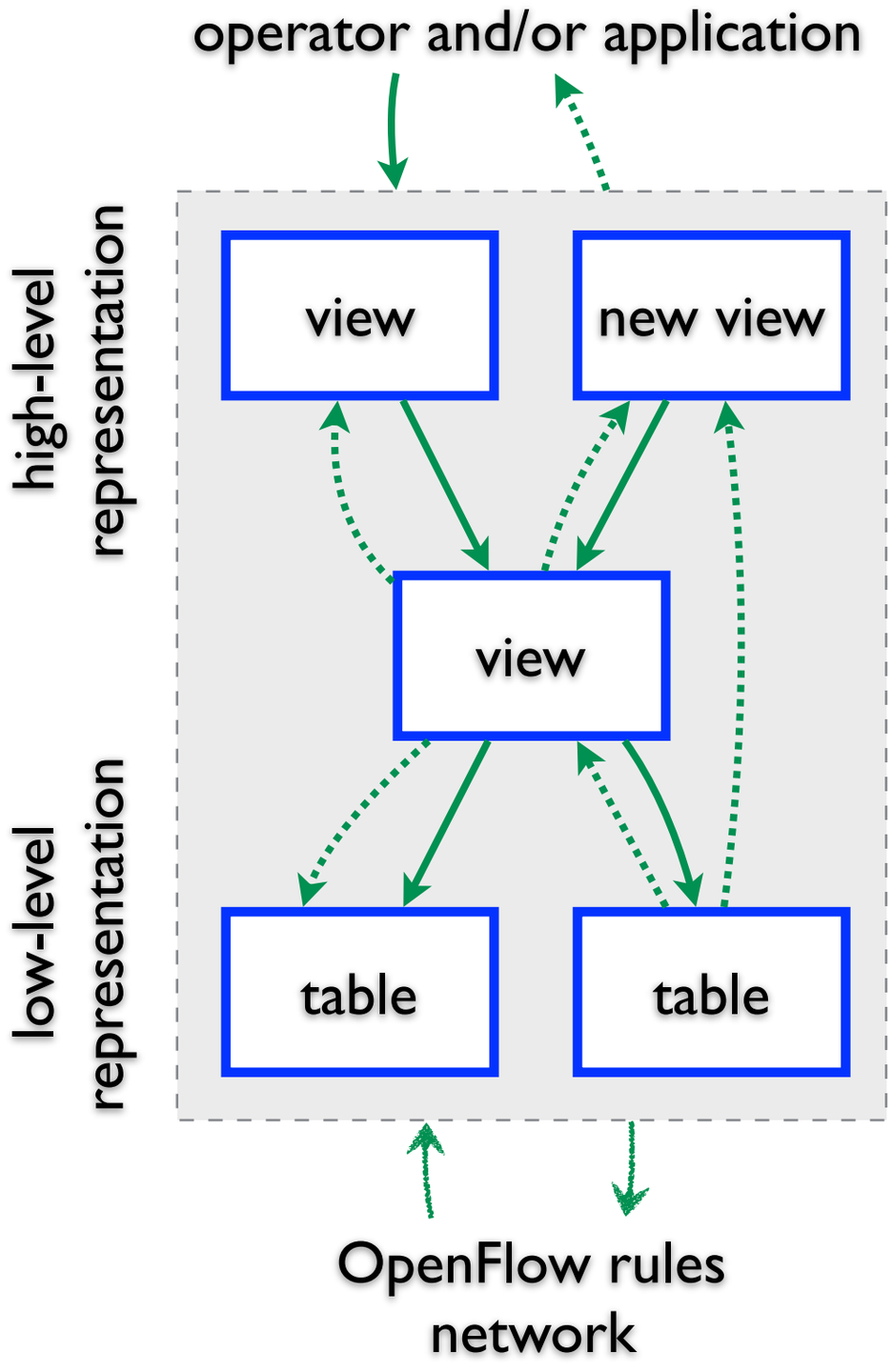
- discard specialized, pre-compiled, fixed structure
- adopt a *plain data representation*
- use a *universal data language*

a database-defined network



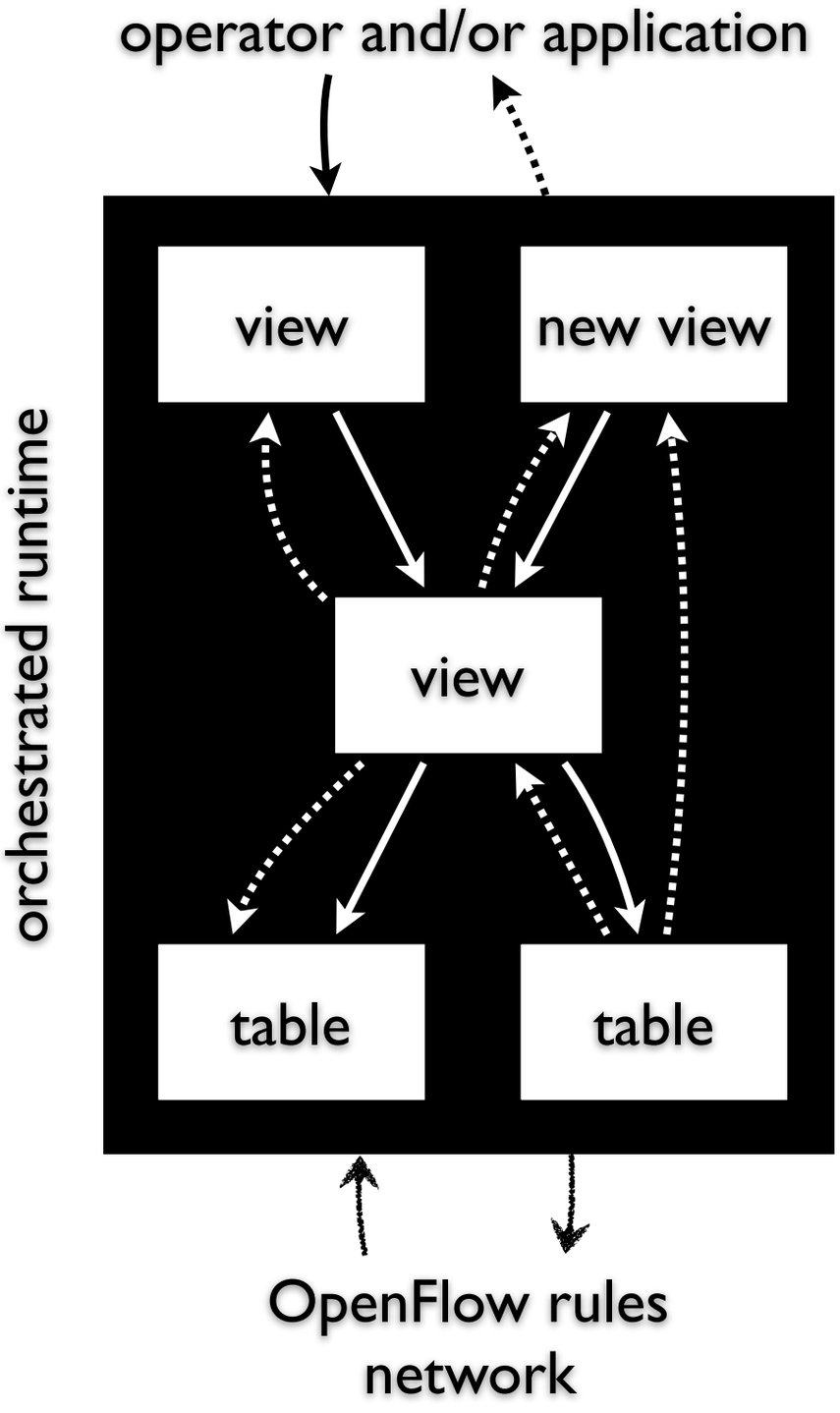
- relation — the plain data representation
  - table — stored relation
  - view — virtual relation

a database-defined network



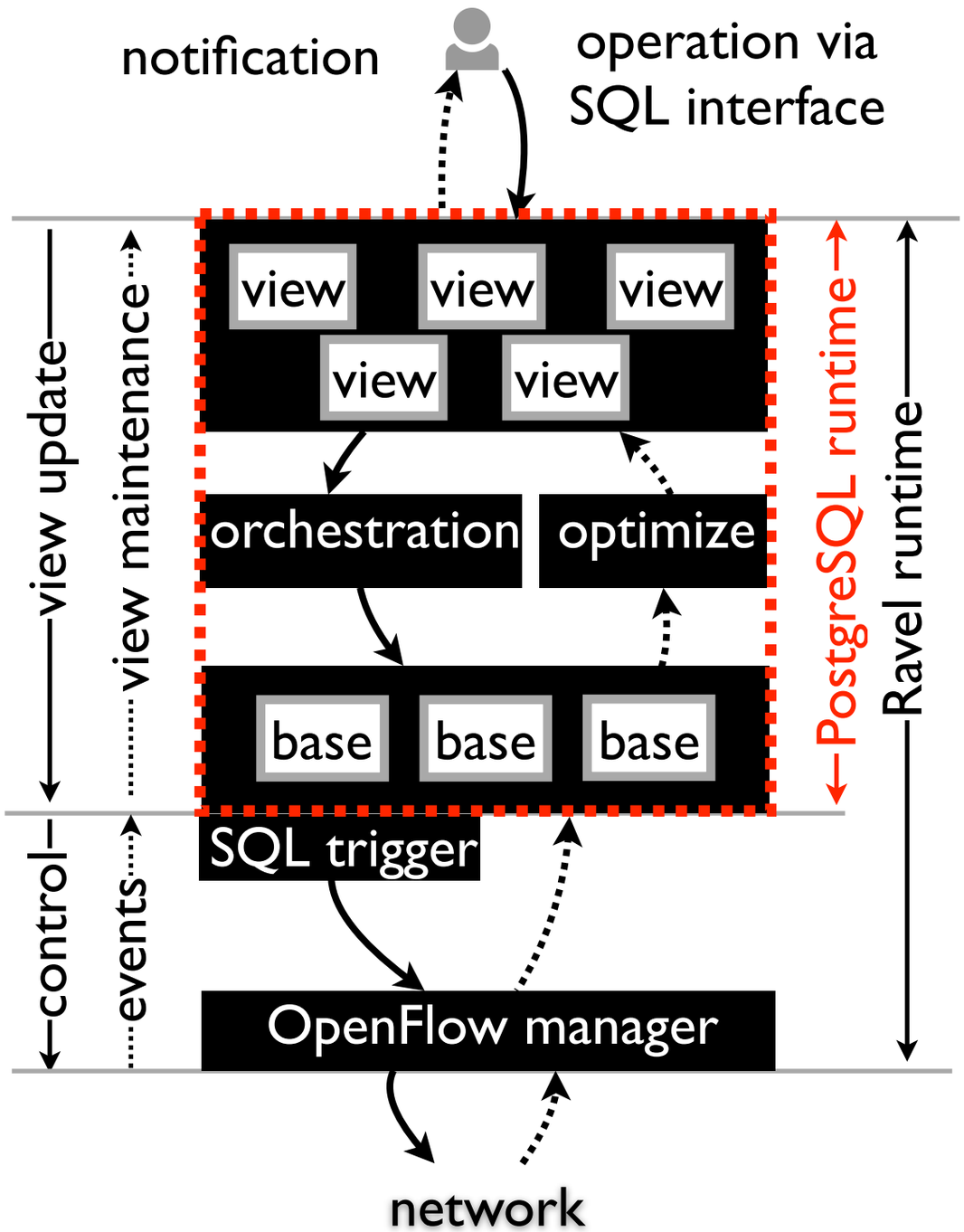
- relation — the plain data representation
  - table — stored relation
  - view — virtual relation
- SQL — the universal data language
  - SQL query .....
  - SQL update ——
  - SQL trigger - - - - -

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

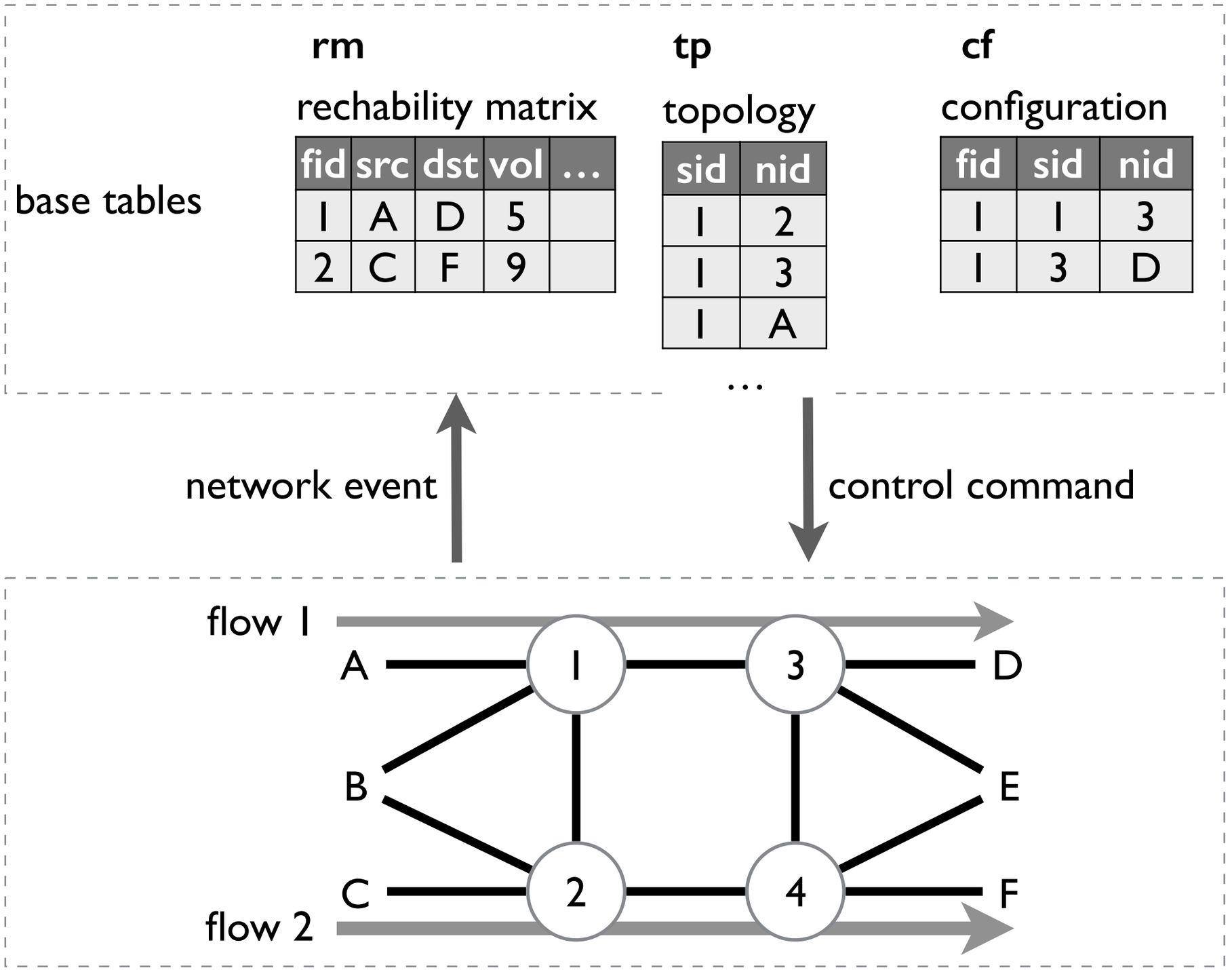
Ravel: a realization with SQL database



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

stateful firewall: allow inbound traffic only when initiated

firewall tables
FW: access control list
FW_user: subnet

```
-- firewall state
CREATE TABLE FW (
  end1 integer, end2 integer, allow integer
);

CREATE TABLE FW_user (uid integer);
```

policy violation view

```
-- firewall constraints
CREATE OR REPLACE VIEW FW_violation AS (
   SELECT fid
   FROM tm
   WHERE FW = 1  AND
      (src, dst) NOT IN
      (SELECT end1, end2 FROM FW_policy_acl)
);
```

rule for violation repair

```
CREATE OR REPLACE RULE FW_repair AS
   ON DELETE TO FW_violation
   DO INSTEAD
         DELETE FROM tm WHERE fid = OLD.fid;
```

two firewall rules
that encode the state transitions

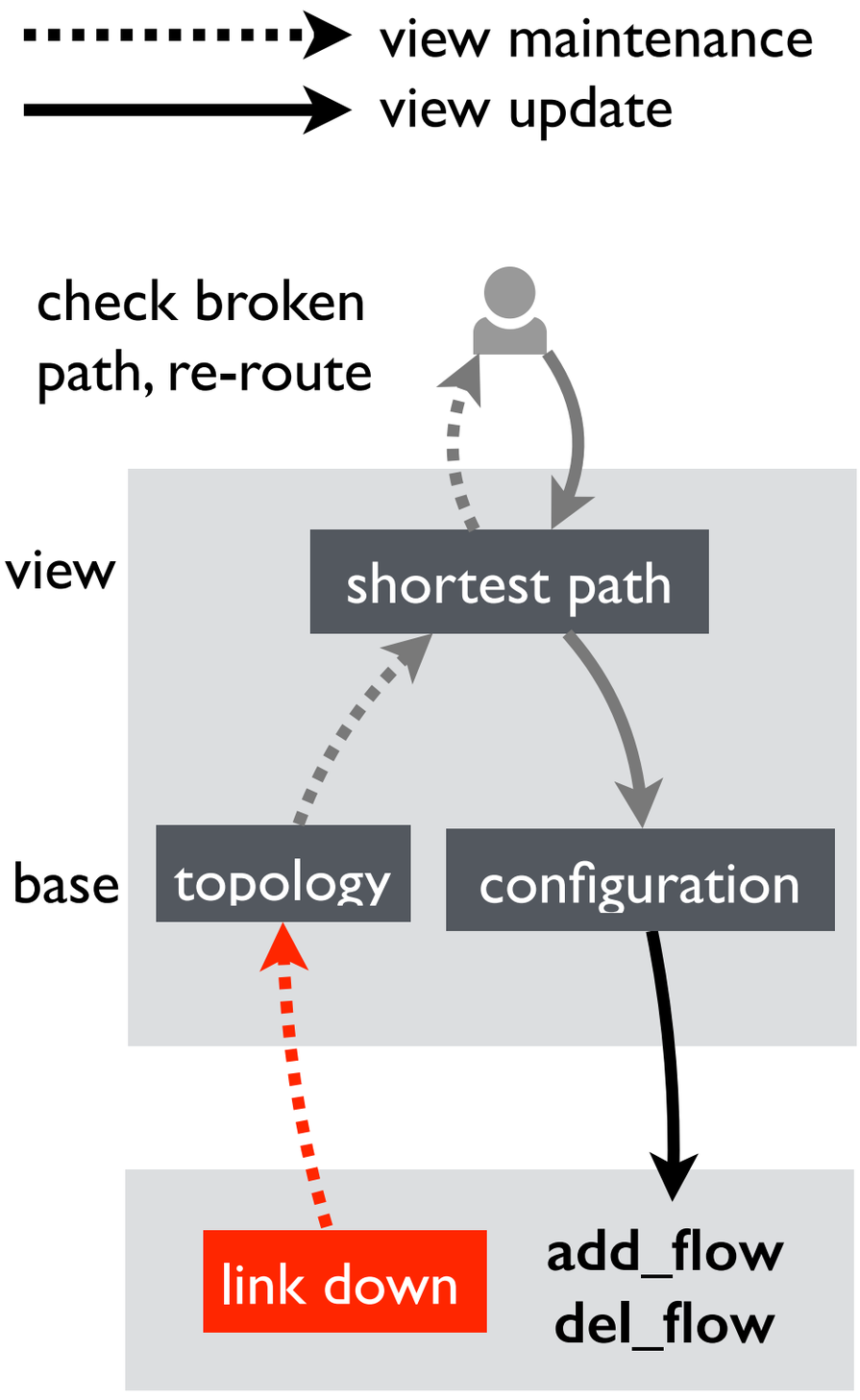
add allowed inbound routes

```
-- firewall state transitions
CREATE RULE FW1 AS
  ON INSERT TO tm
  WHERE ((NEW.src, NEW.dst) NOT IN
     (SELECT end2, end1 FROM FW)) AND
      (NEW.src IN (SELECT * FROM FW_user))
  DO ALSO (
     INSERT INTO FW
       VALUES (NEW.dst, NEW.src, 1););
```

remove outdated inbound routes

```
CREATE RULE FW2 AS
  ON DELETE TO tm
  WHERE (SELECT count(*)
         FROM tm WHERE src = OLD.src AND
                  dst = OLD.dst) = 1
  DO ALSO
     DELETE FROM FW
     WHERE end2 = OLD.src
           AND end1 = OLD.dst;
```

orchestration across representations



shortest path rule:
upon broken path, re-route

	shortest path	
	…	path
-		{…,172,39,156,…}
+		{…,172,38,148,…}

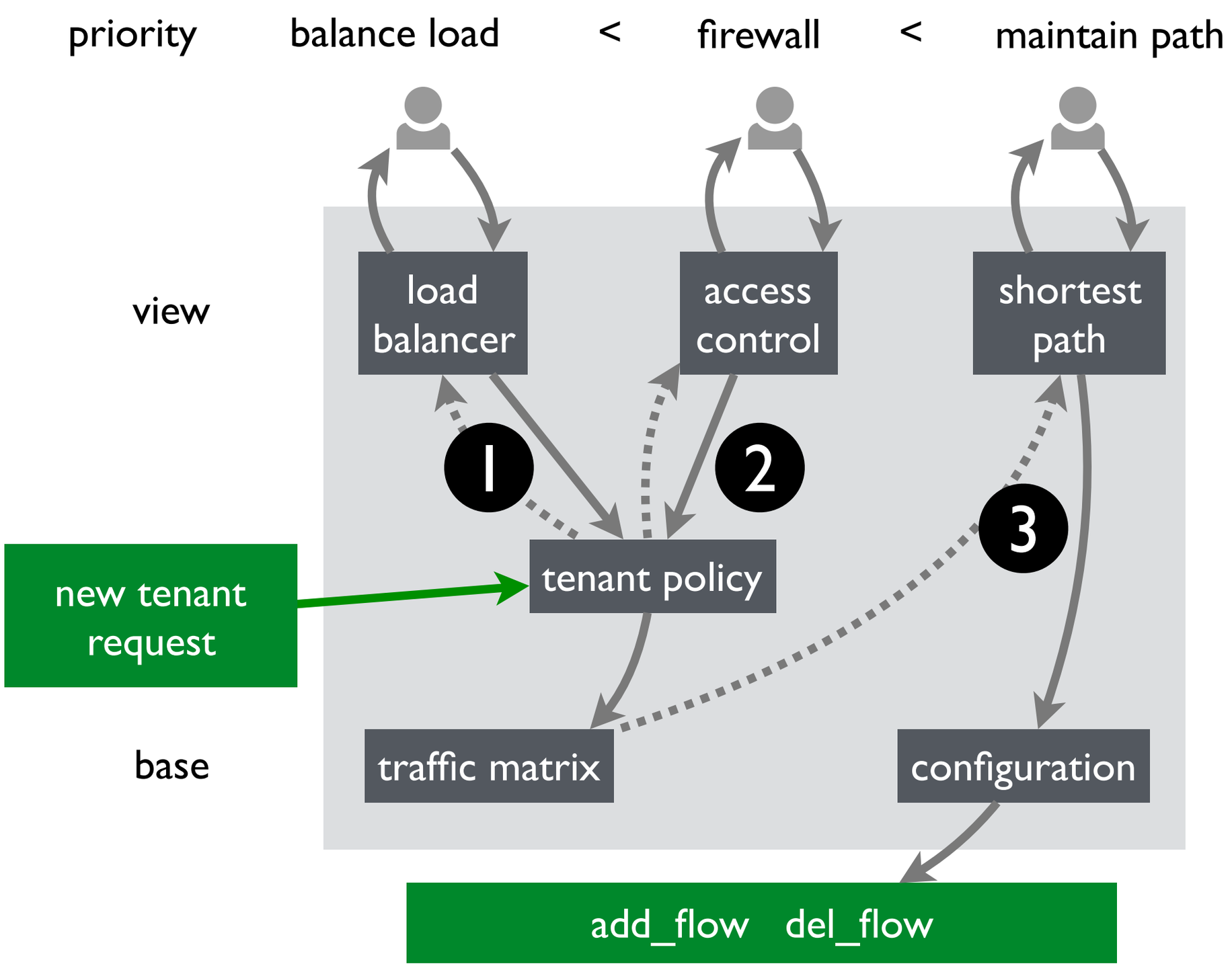
	topology		
	sid	nid	active
-	172	39	1
+	172	39	0

	configuration		
	fid	sid	nid
-	…	172	39
-	…	39	156
+	…	172	38
+	…	38	148

Mininet link (172,39) down

orchestrated updates: re route via (172, 38)

orchestration across applications



orchestration across applications

rebalance route: move 1 unit from 1003 to 1034

	load balancer	
	sid	load
-	1003	4
+	1003	3
-	1034	1
+	1034	2

	access control		
	src	dst	allow
+	1238	1034	1

	shortest path			
	…	src	dst	path
+	…	1238	1034	{1238,…,1034}

+(1238,1034)

1 +(1003,4)

2

+(1238,1034,1)

3

+(…,1238,…)
…
+(…,…, 1034)

	tenant policy		
	…	host	server
+	…	1238	1003
+		1238	1034

+(…,1238,1003)

tenant requests host 1238 to server 1003

+(…,1238,1034)

+(…,1238,1034, path=?)

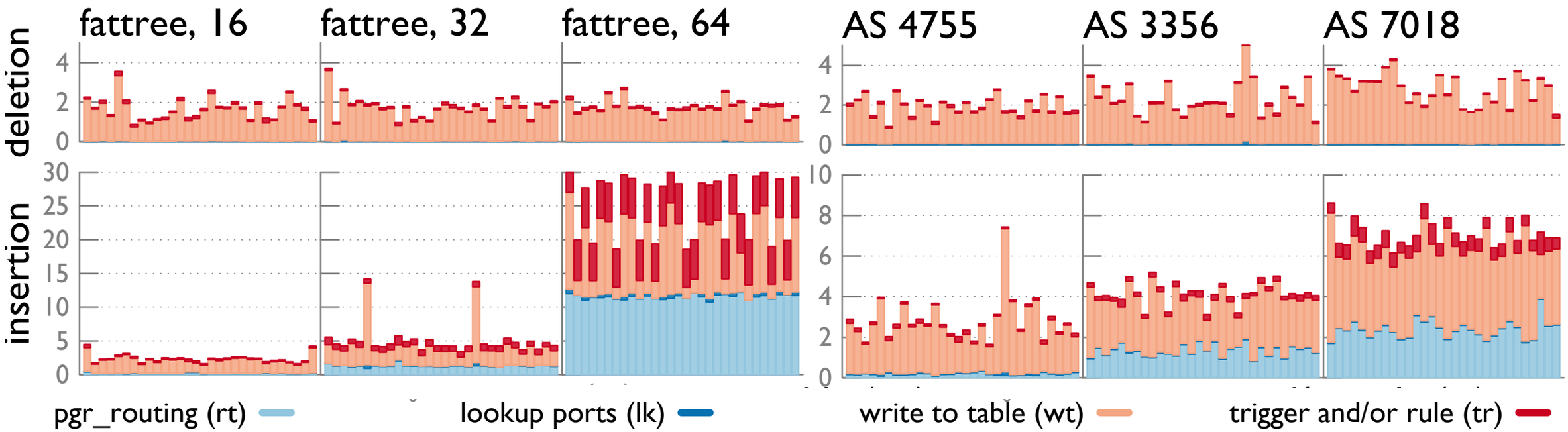
	traffic matrix		
	fid	sid	nid
+	…	1238	1034

	configuration		
	fid	sid	nid
+	…	…	1034

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

evaluation

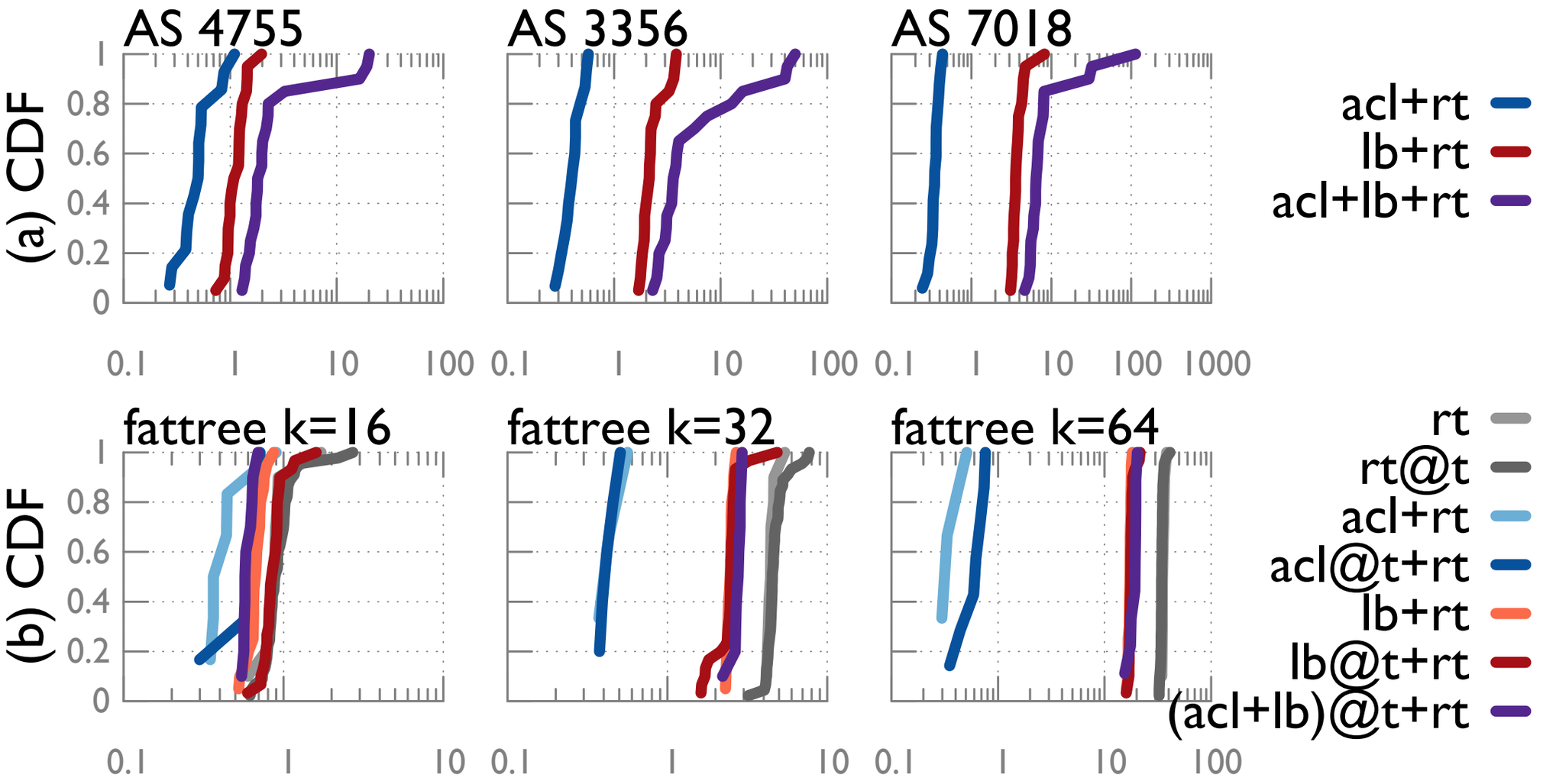
profiling database delay — route insertion/deletion



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

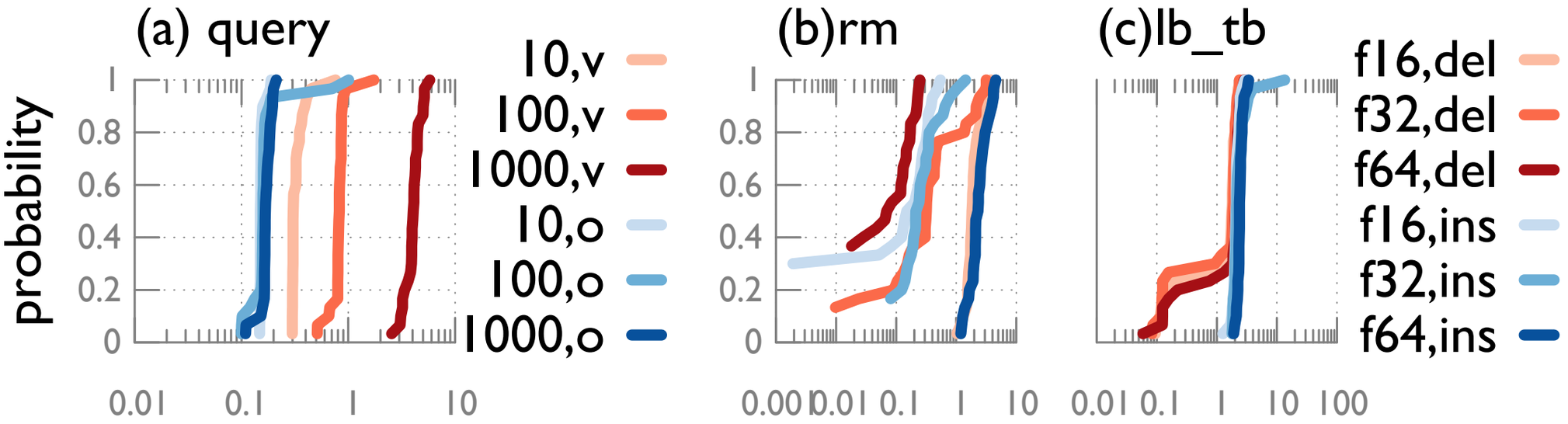
orchestrating access control(acl), load balancer(lb), and routing(rt): normalized per-rule delay (ms)



evaluation

optimizing application—materializing views

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



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, interoperability (IXP)
  - distributed and federated database

ongoing work

- bootstrapping database
- 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