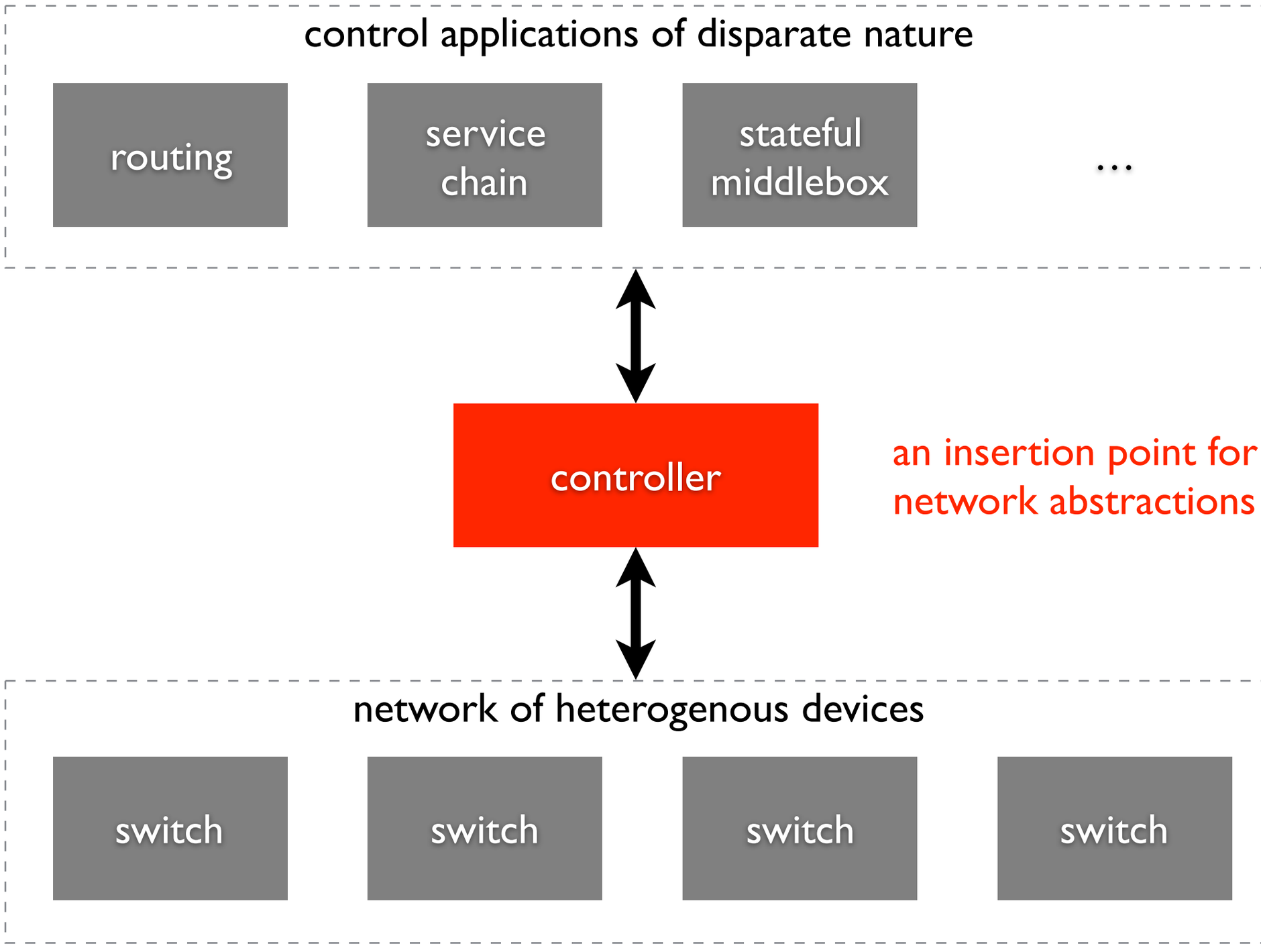

# *Ravel:* a database-defined network

Anduo Wang* Xueyuan Mei† Jason Croft†
Matthew Caesar† Brighten Godfrey†

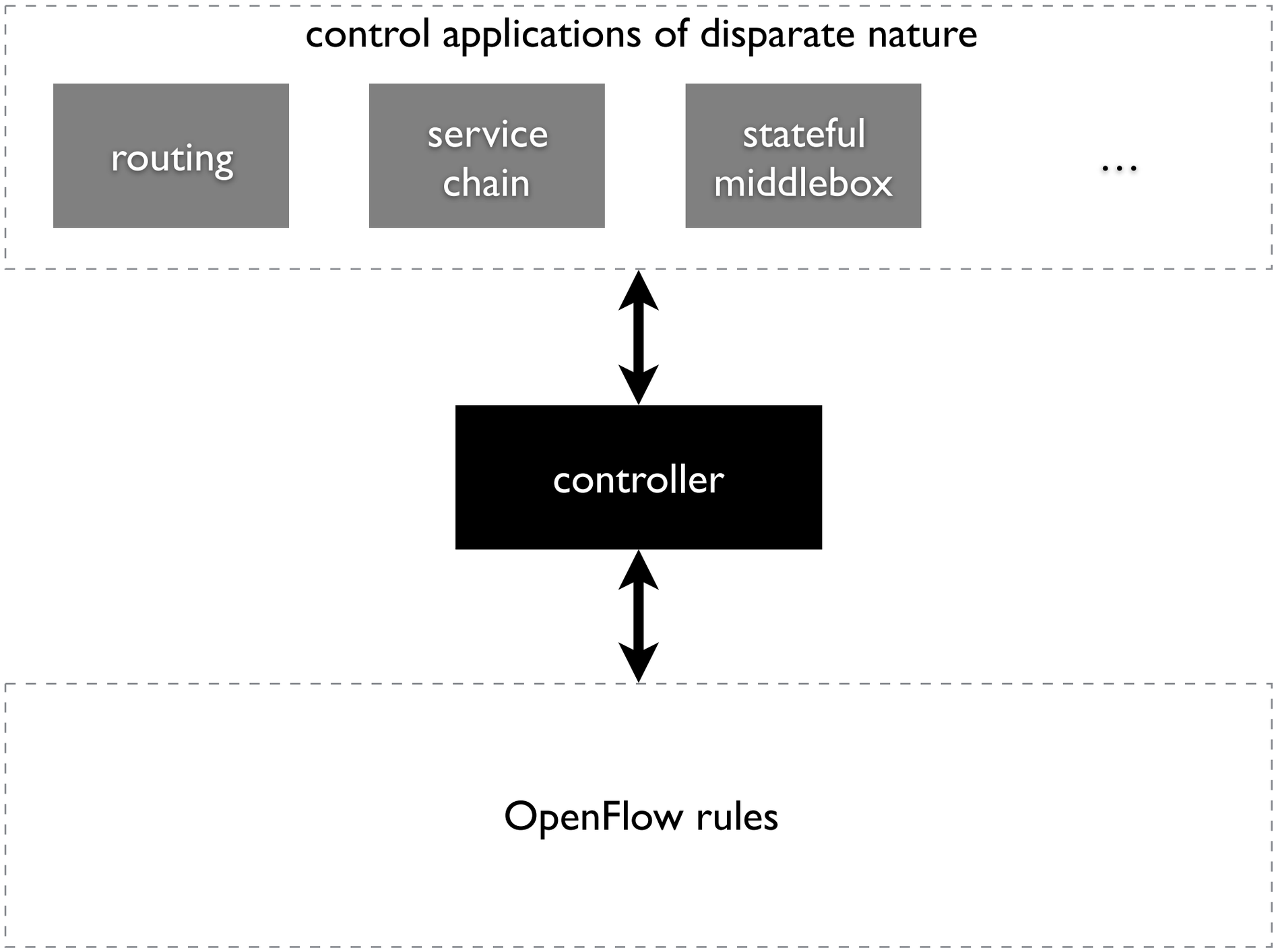
**Temple University* †*University of Illinois Urbana-Champaign*

March 14, 2016 SOSR'16

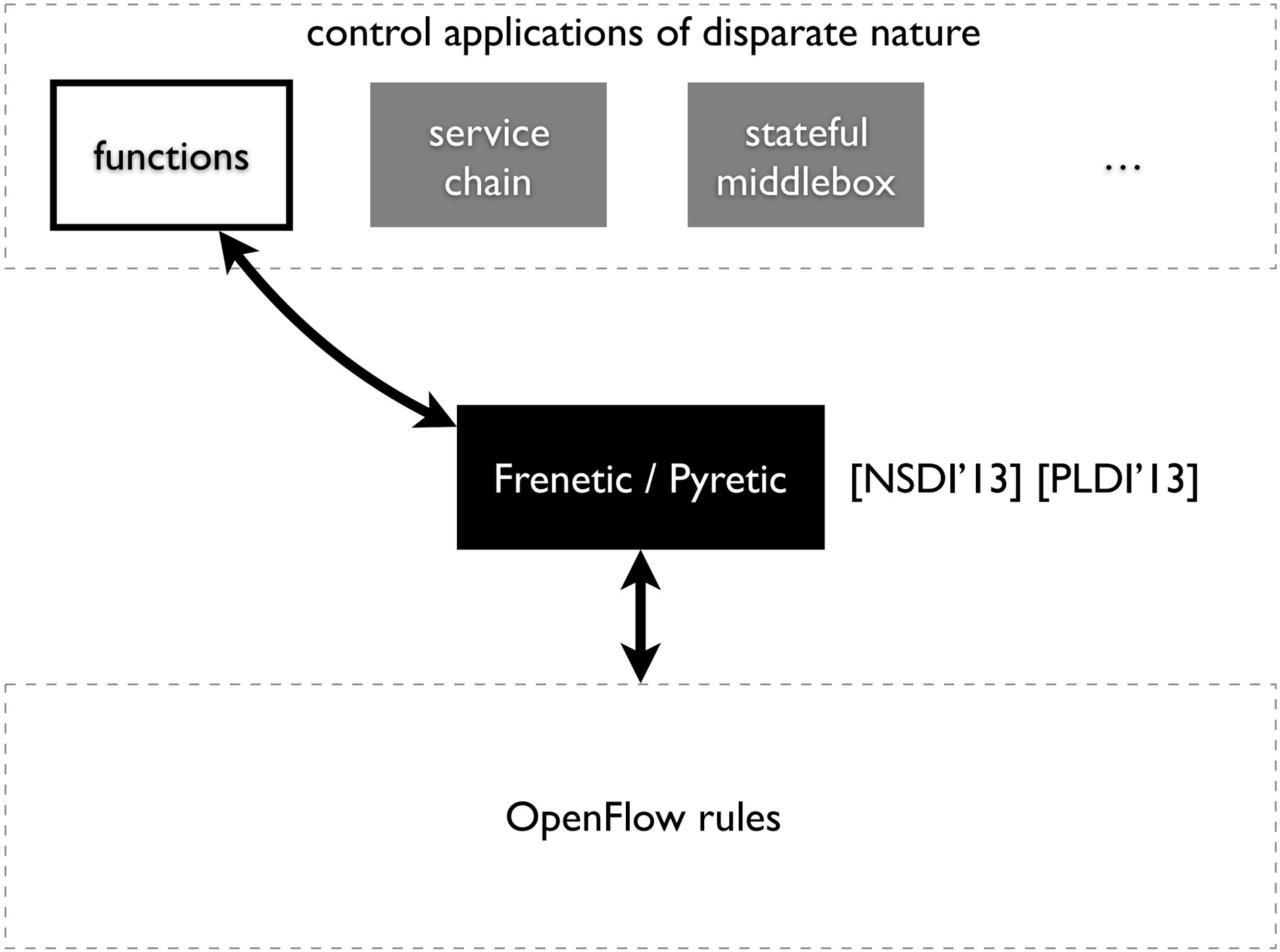
# software-defined network



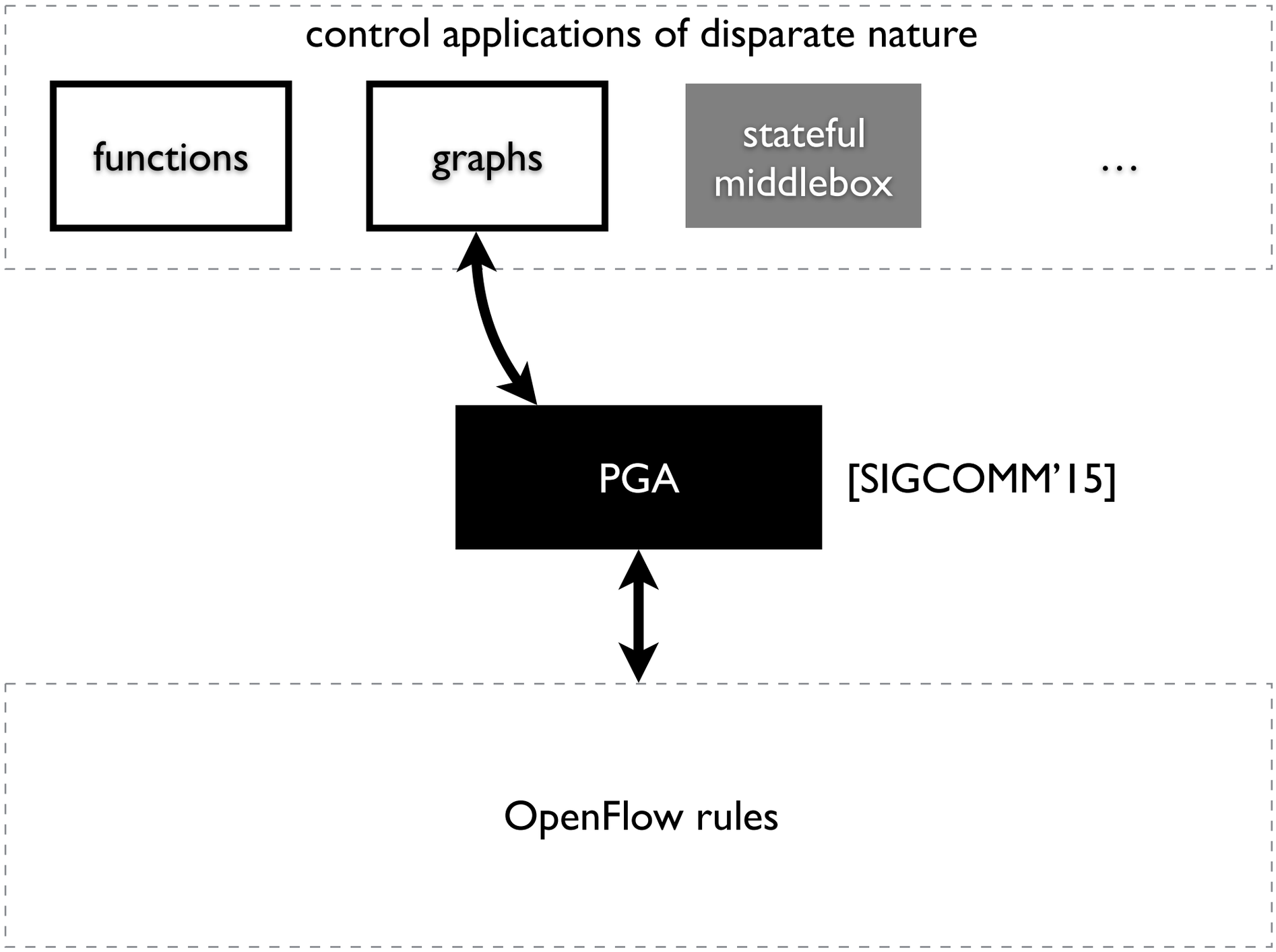
# abstractions



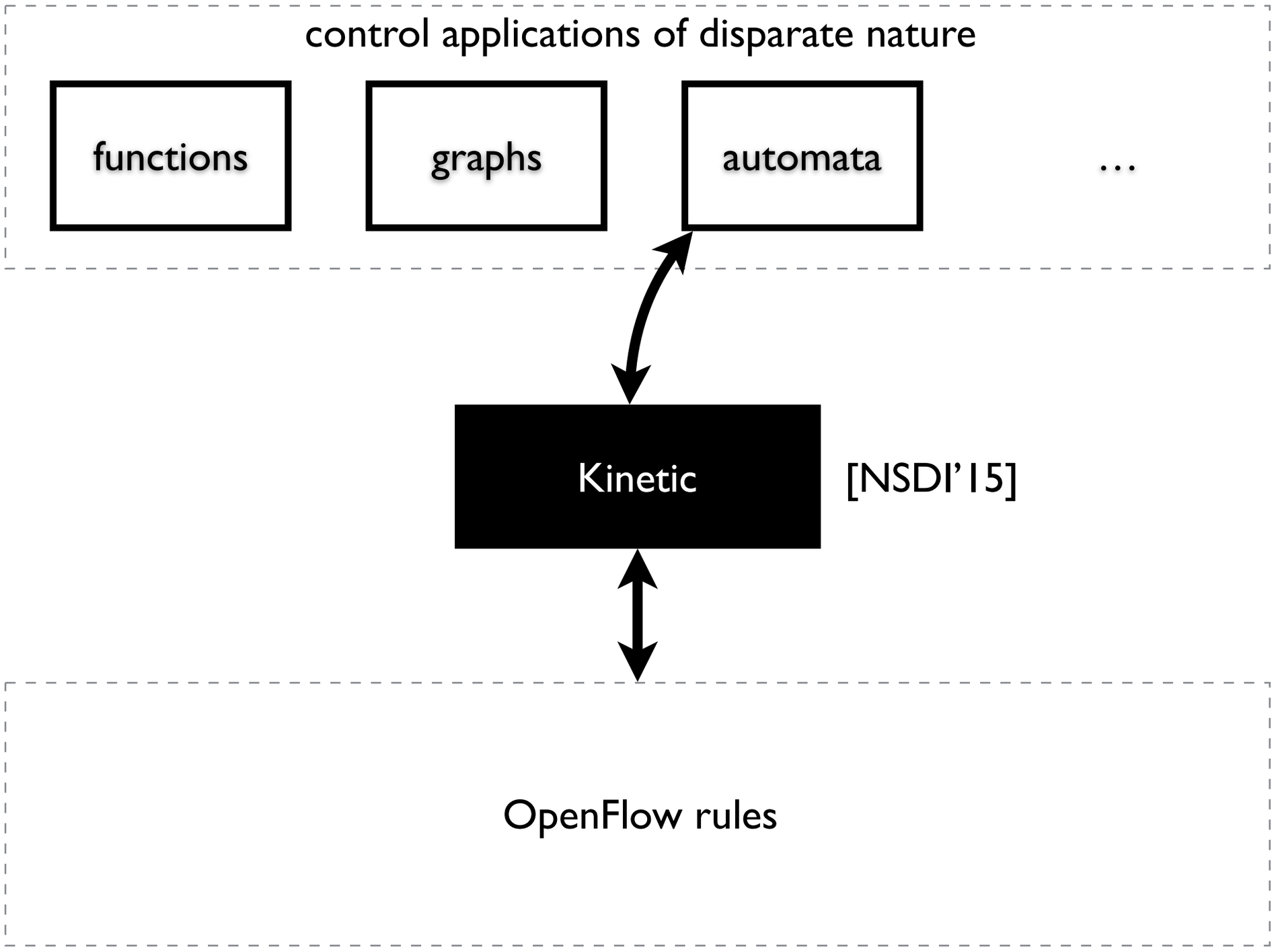
# abstractions



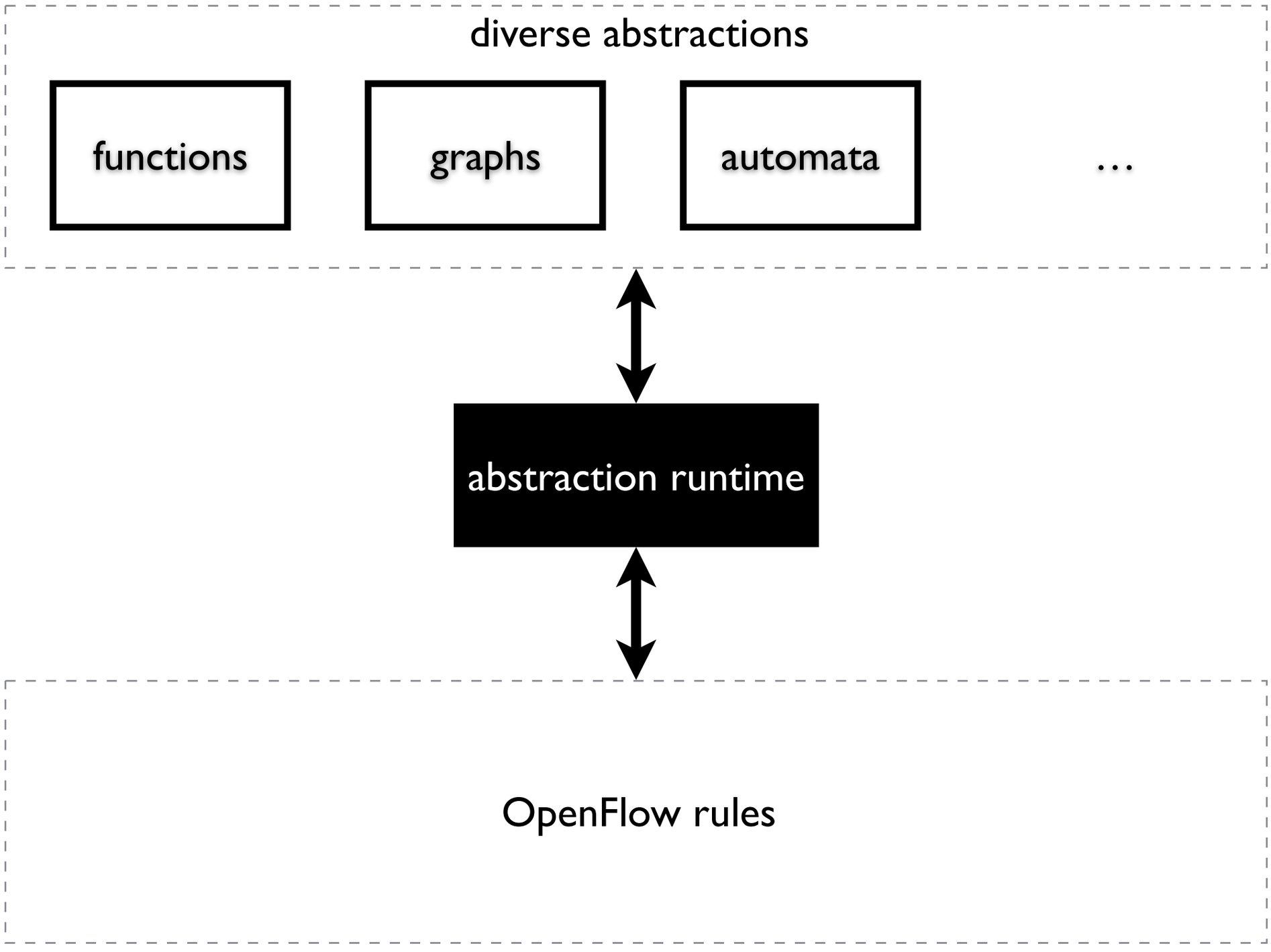
# abstractions



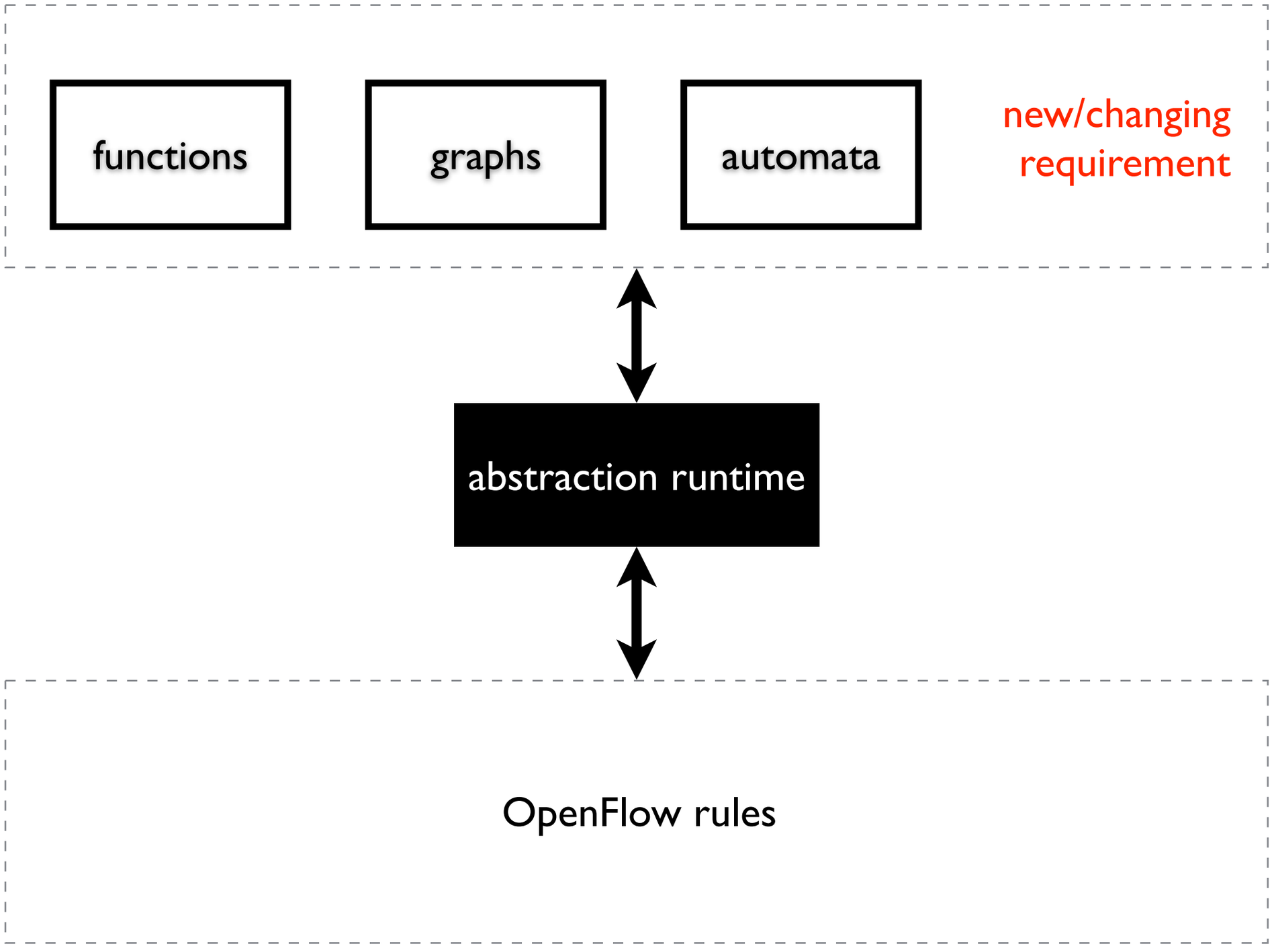
# abstractions



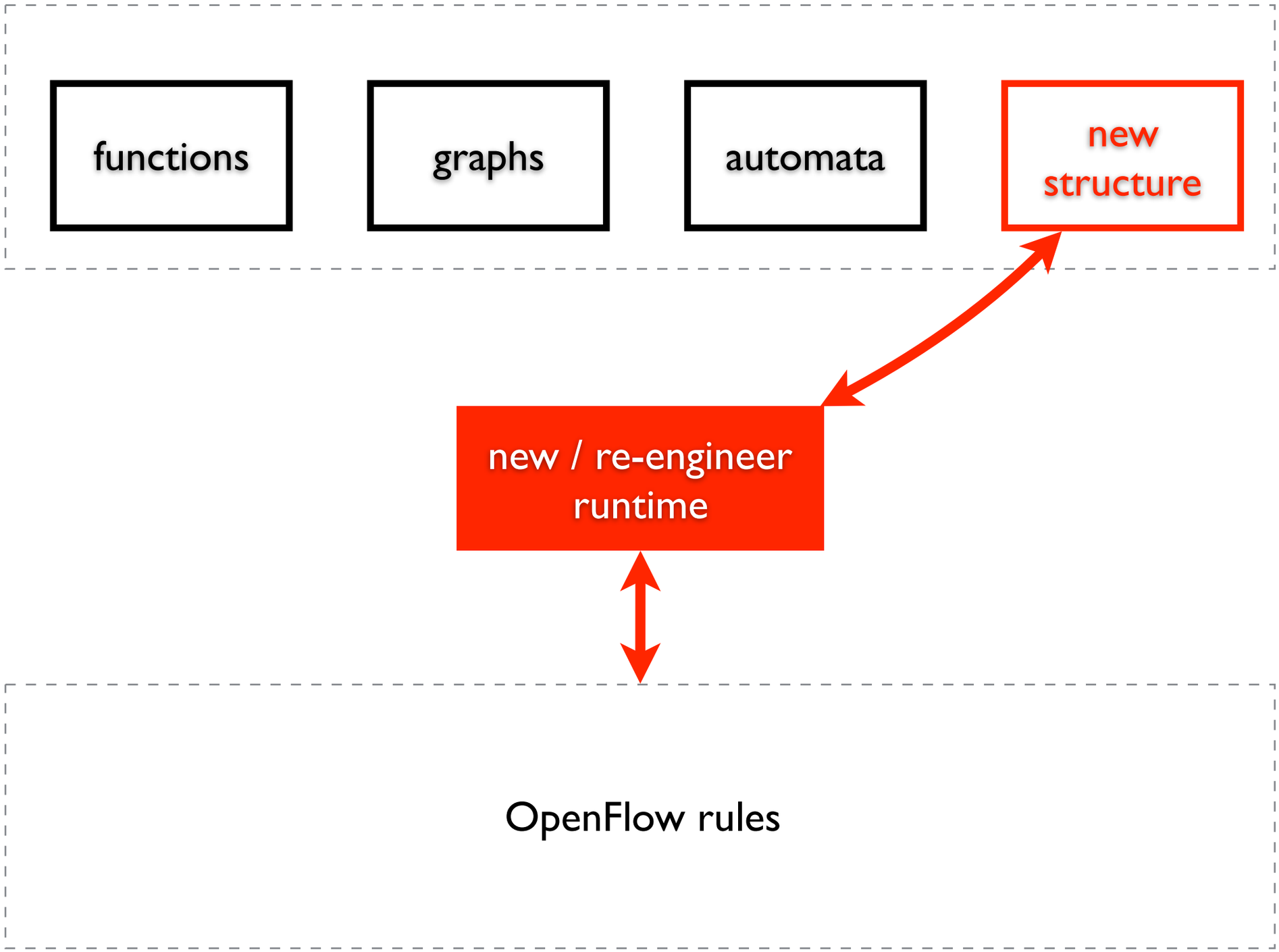
# abstractions



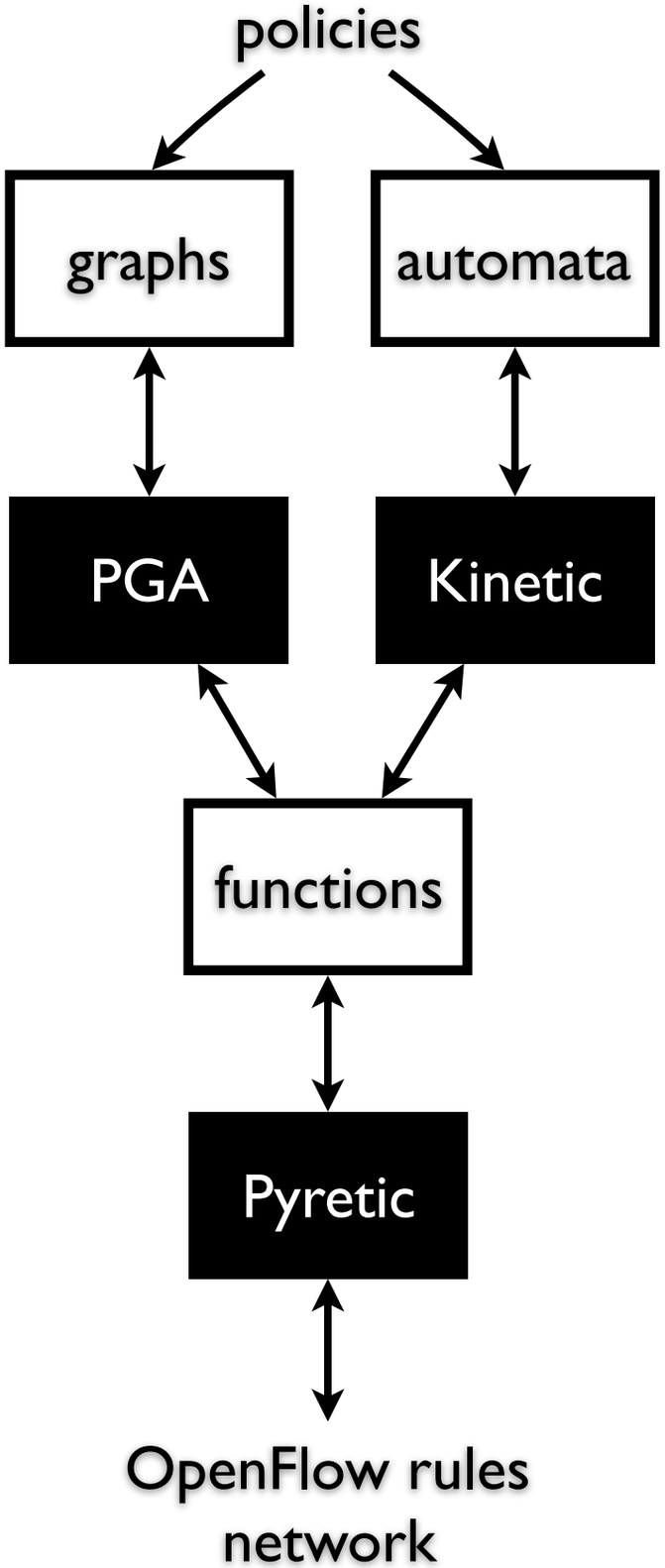
# but network keeps evolving



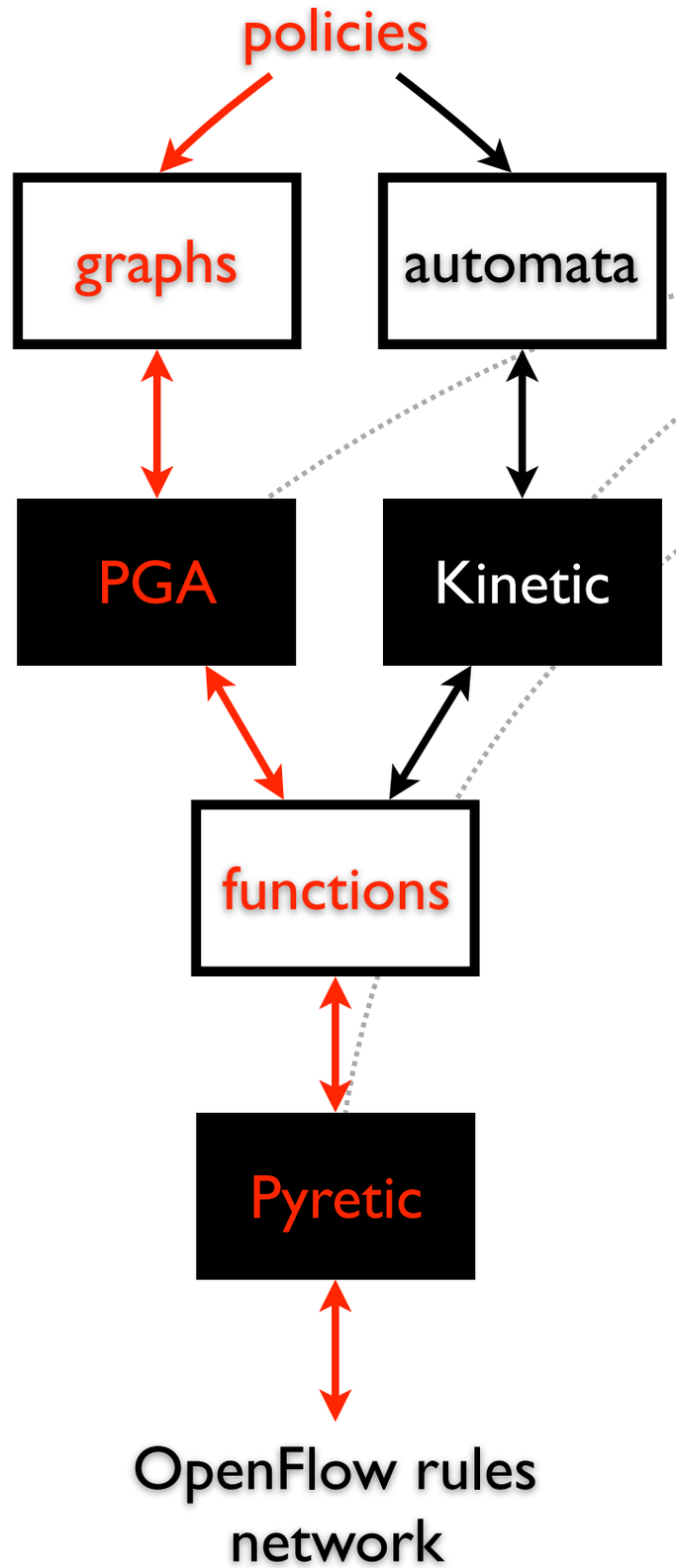
# but network keeps evolving



# and applications (components) interact



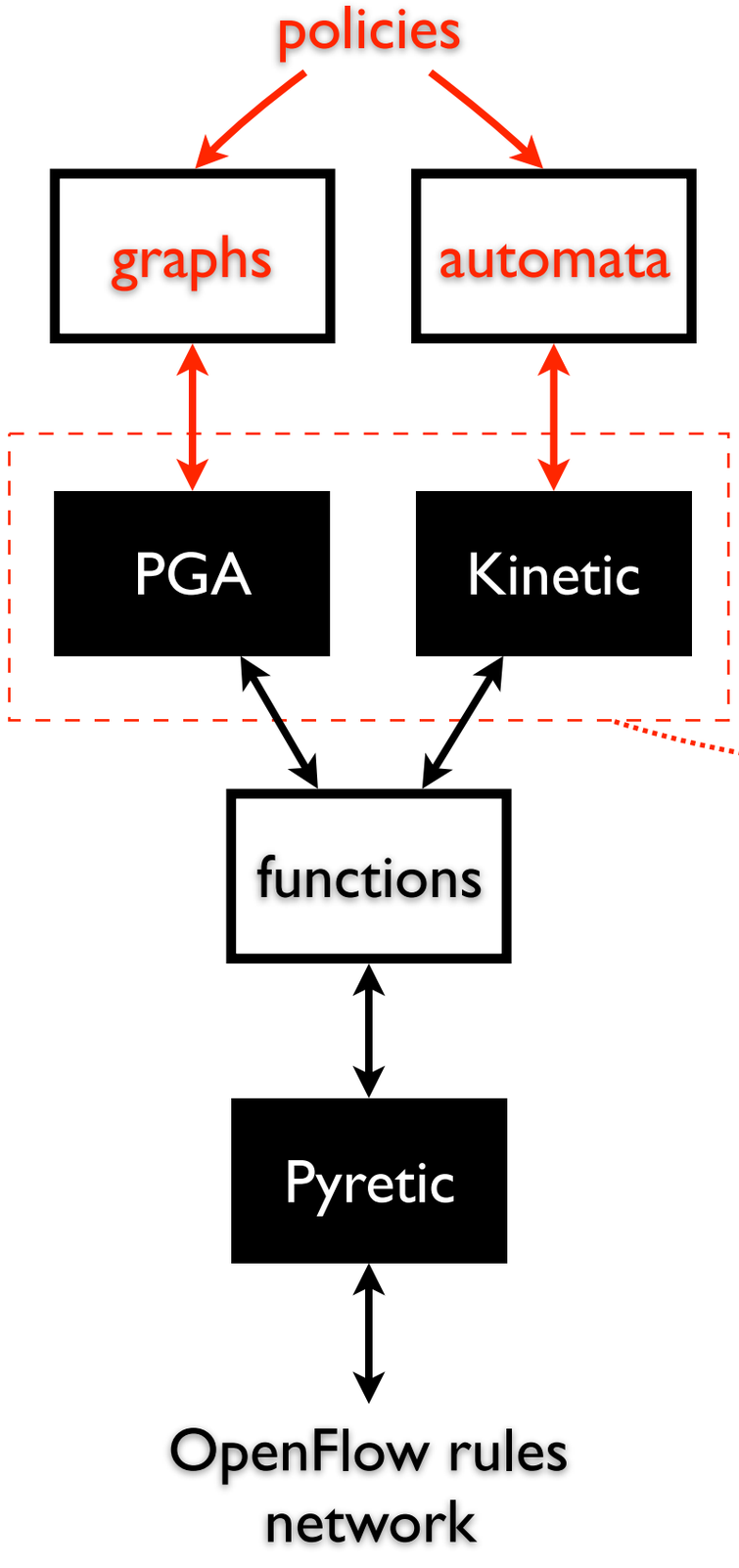
# and applications (components) interact




language-level orchestration restricted to each abstraction

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

# and applications (components) interact

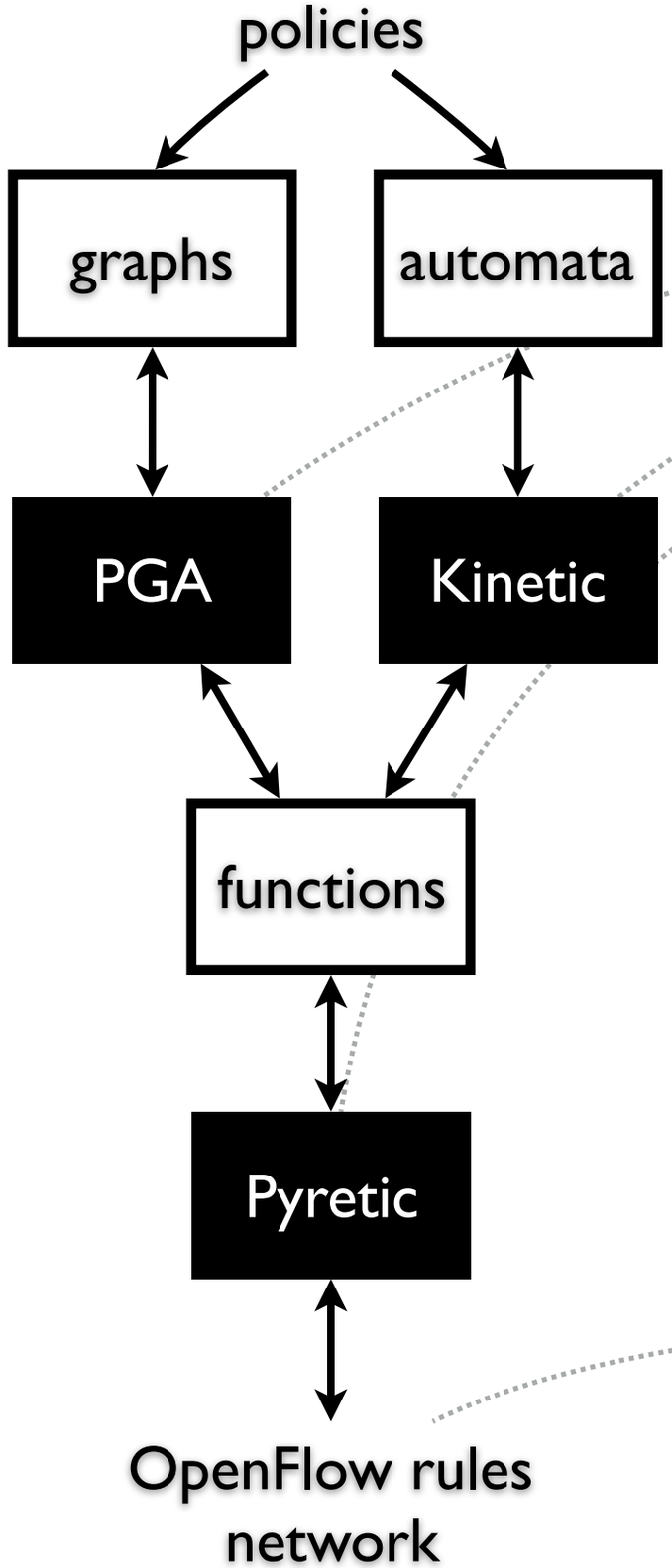



language-level orchestration restricted to each abstraction

composing (+) policy
→ graph +? automata

integrate the runtime, hard-wiring internals?

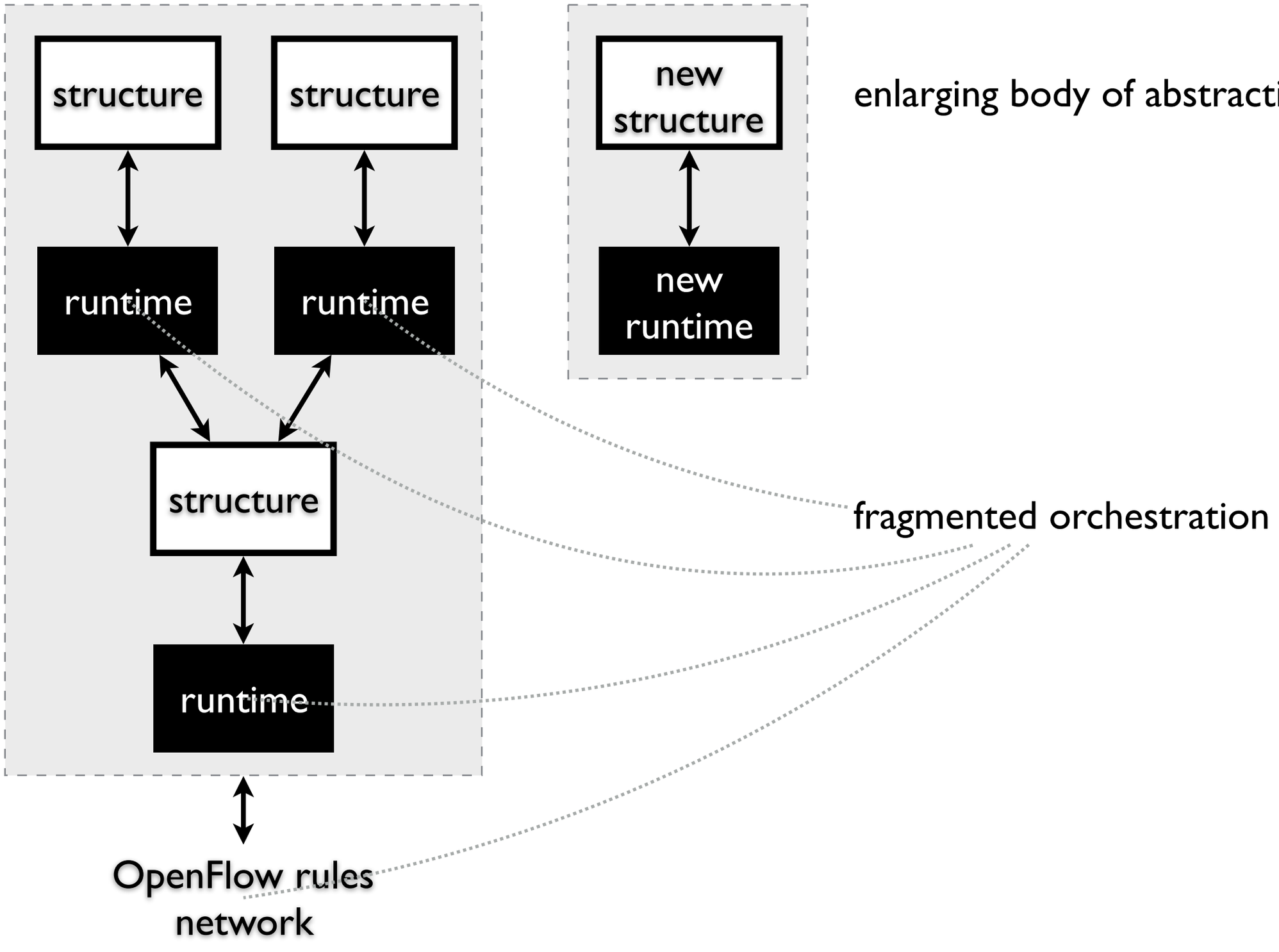
# and applications (components) interact



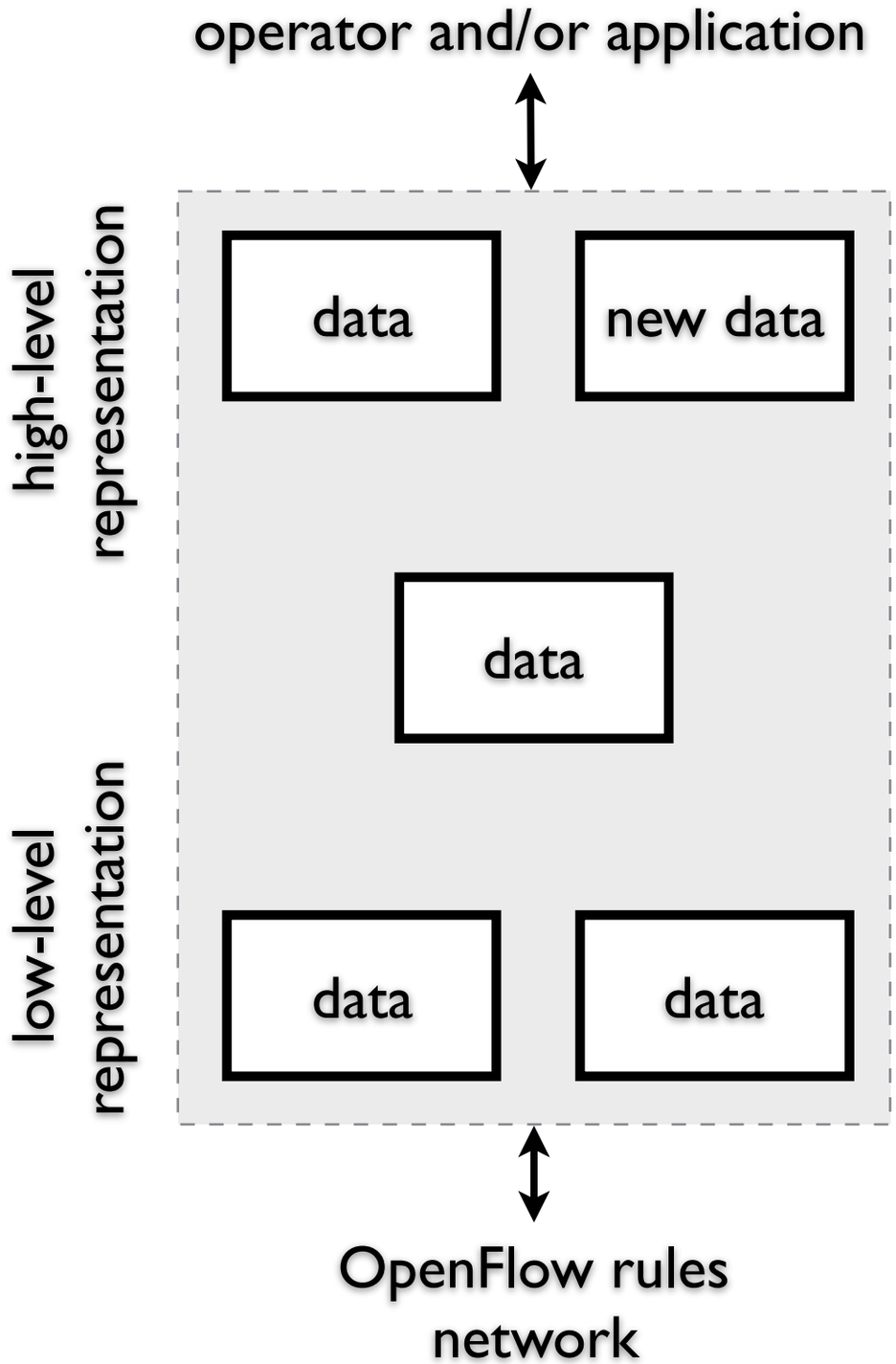

language-level orchestration restricted to each abstraction

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

# current states of abstraction



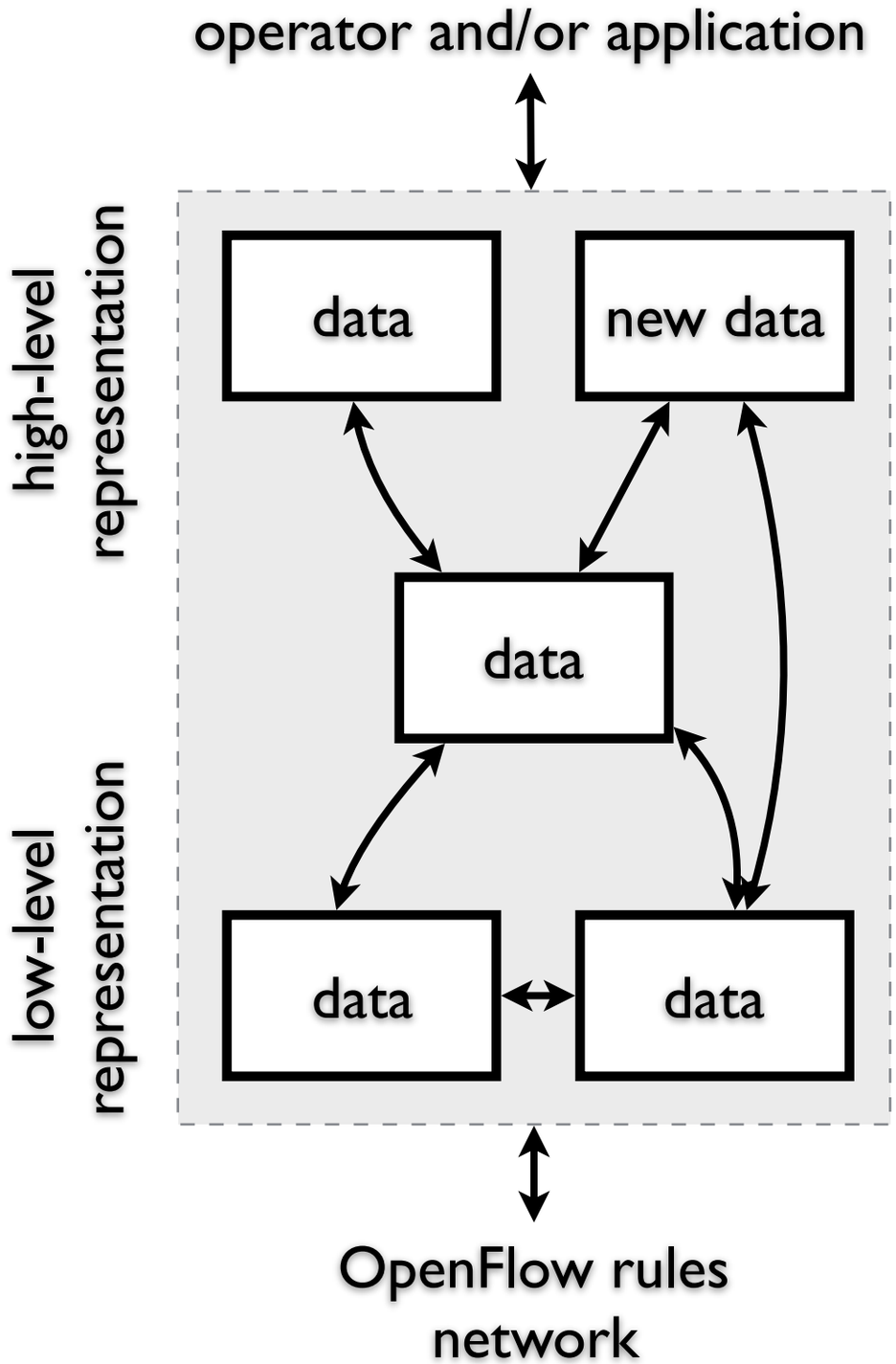
# our perspective




SDN control revolves around data representation

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

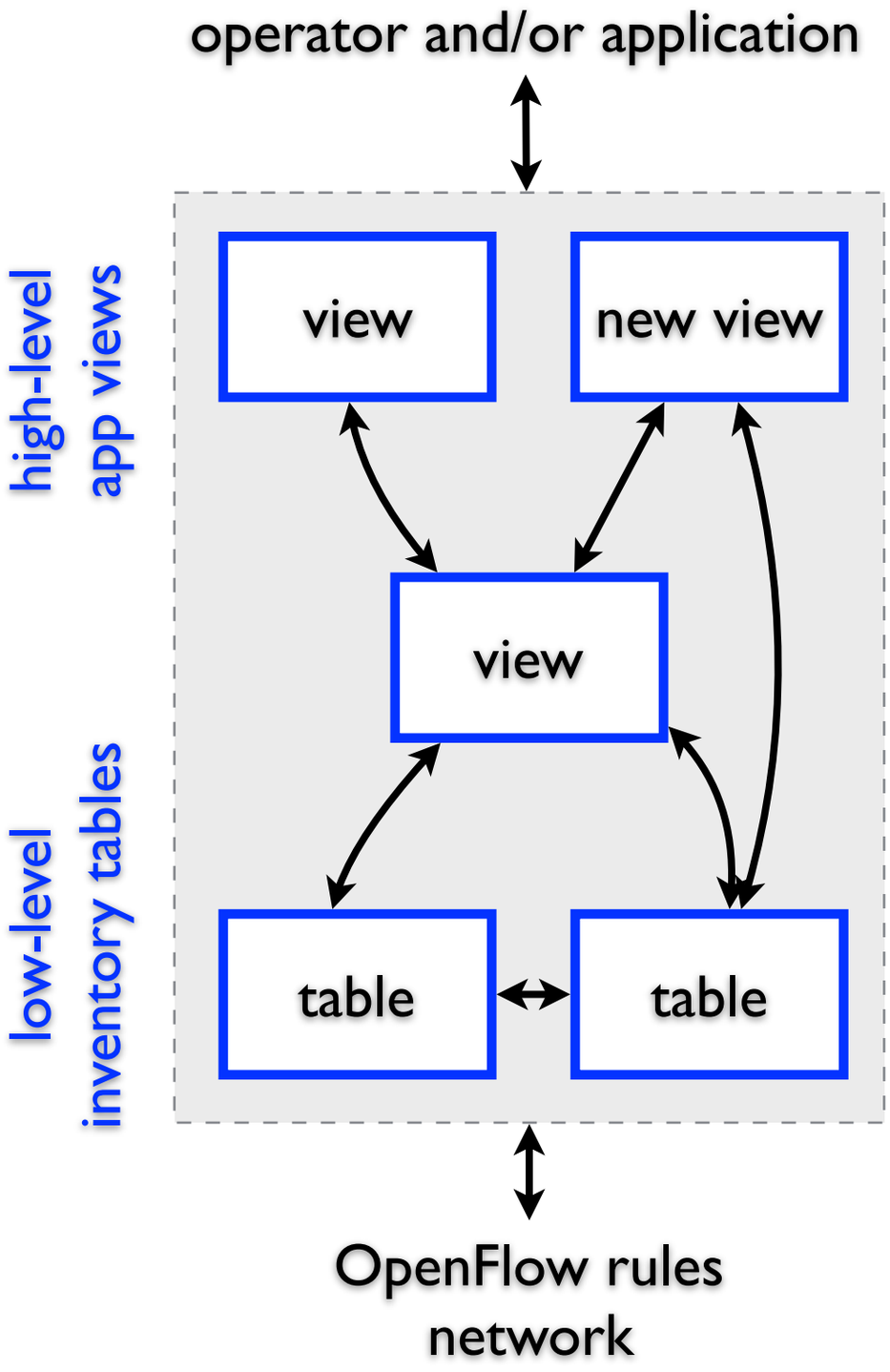
# our perspective




SDN control revolves around data representation

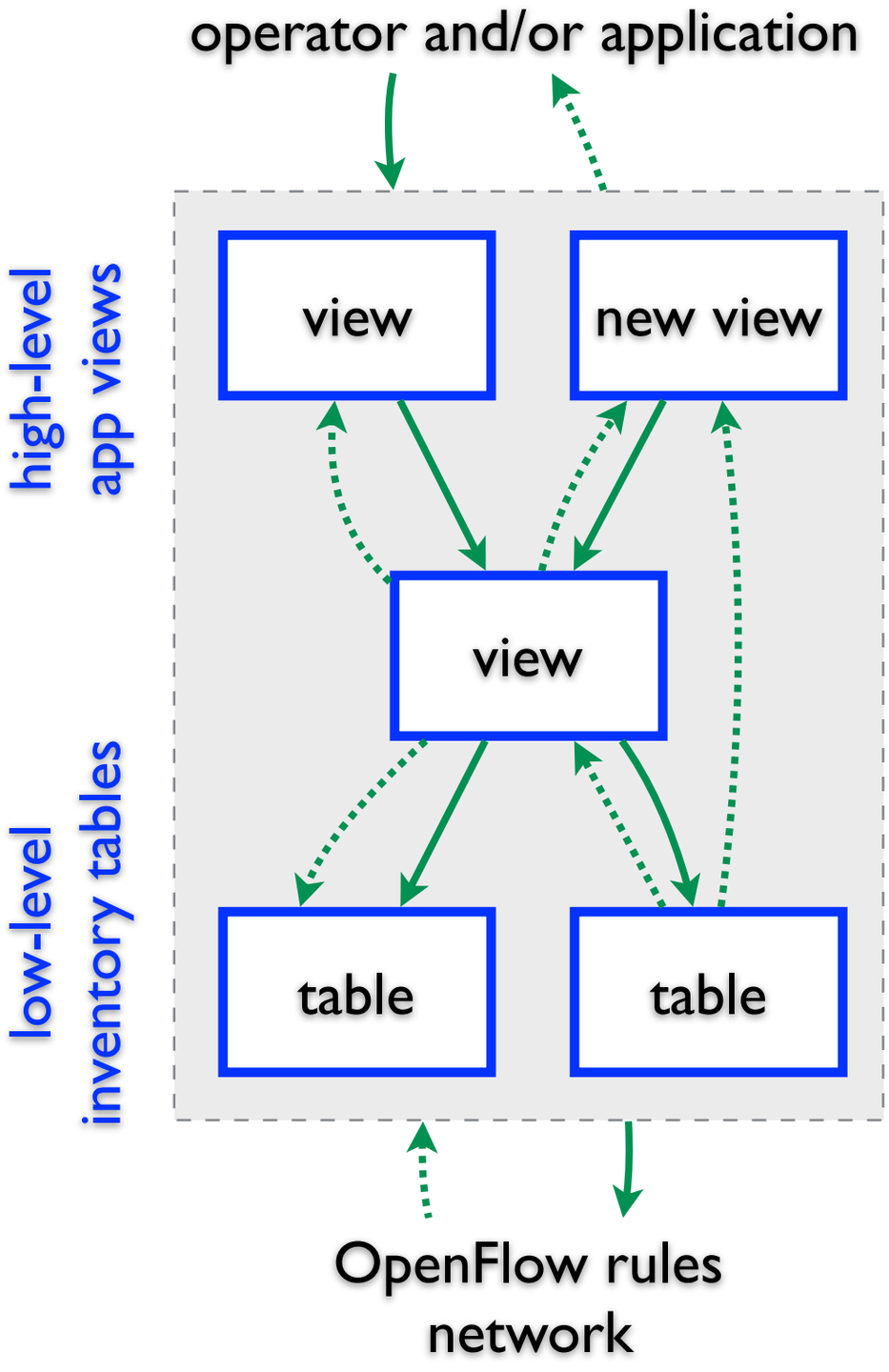
- discard specialized, pre-compiled, fixed structures
- adopt a *plain data representation*
- use a *universal data language*

# a database-defined network



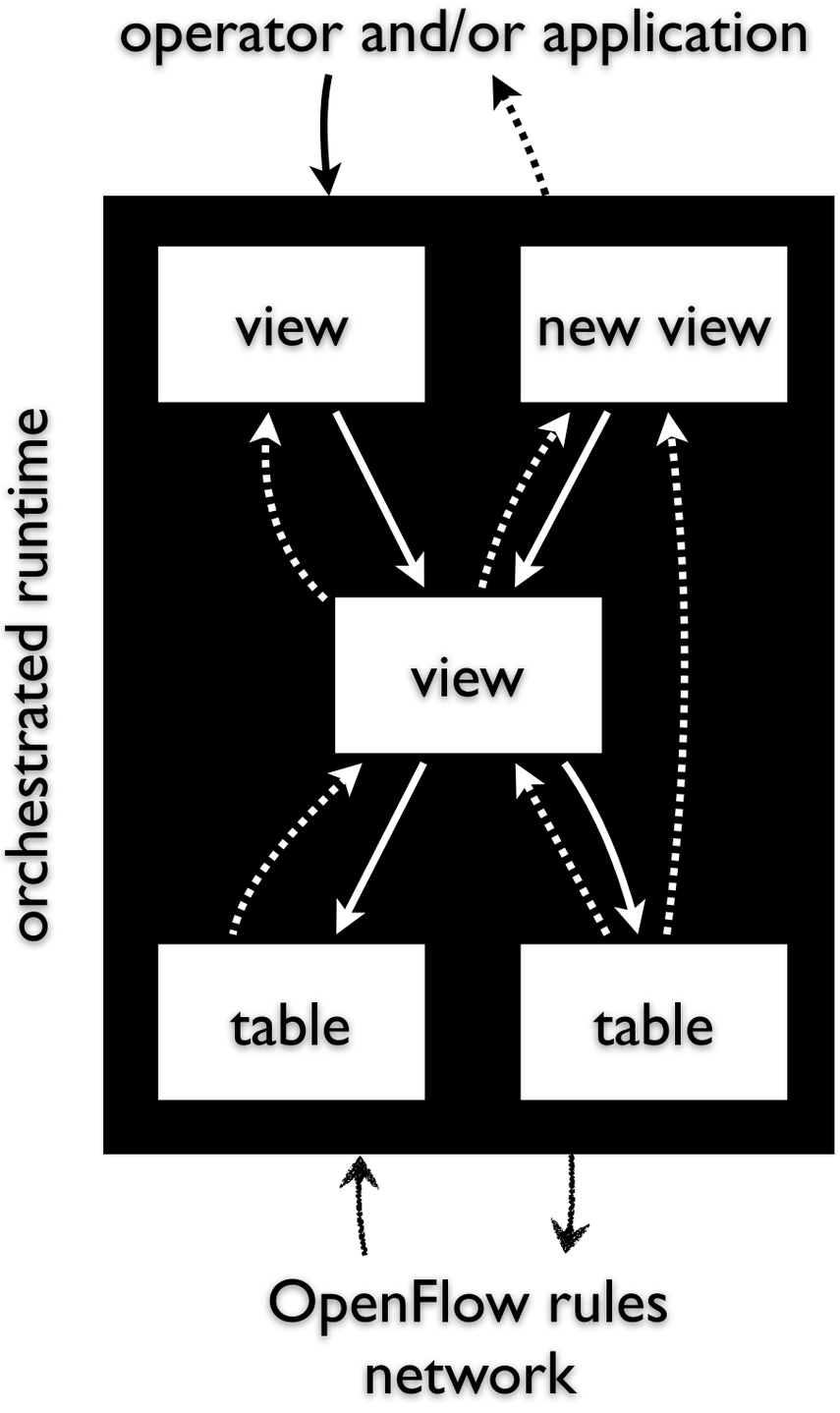

- relation — the plain data representation
  - table — stored relation
  - view — virtual relation

# a database-defined network



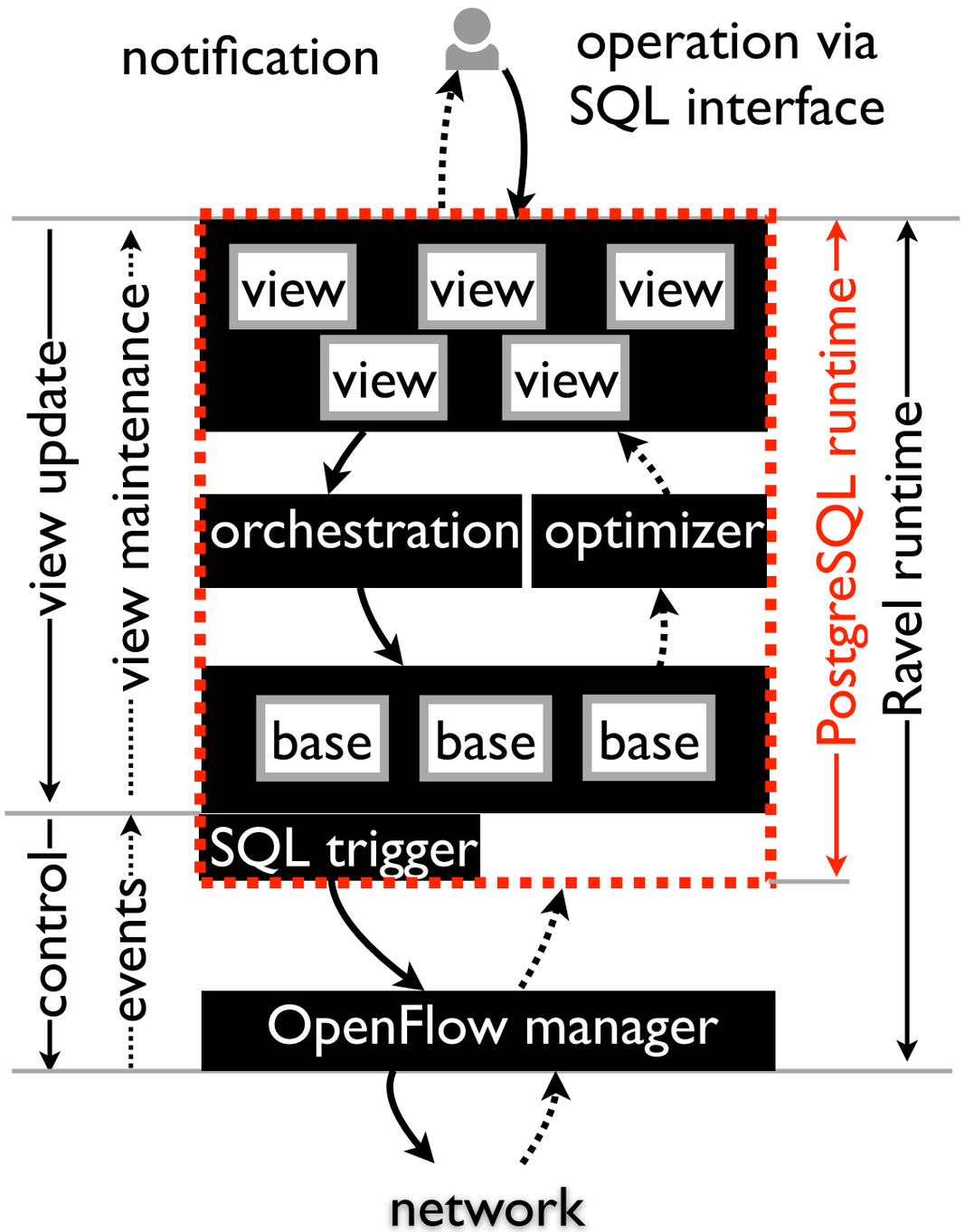

- relation — the plain data representation
  - table — stored relation
  - view — virtual relation
- SQL — the universal data language
  - query, update, trigger, rule

# a database-defined network




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

base tables

**rm**

rechability matrix

| fid | src | dst | vol | … |
|---|---|---|---|---|
| 1 | A | D | 5 | |
| 2 | C | F | 9 | |

**tp**

topology

| sid | nid |
|---|---|
| 1 | 2 |
| 1 | 3 |
| 1 | A |

…

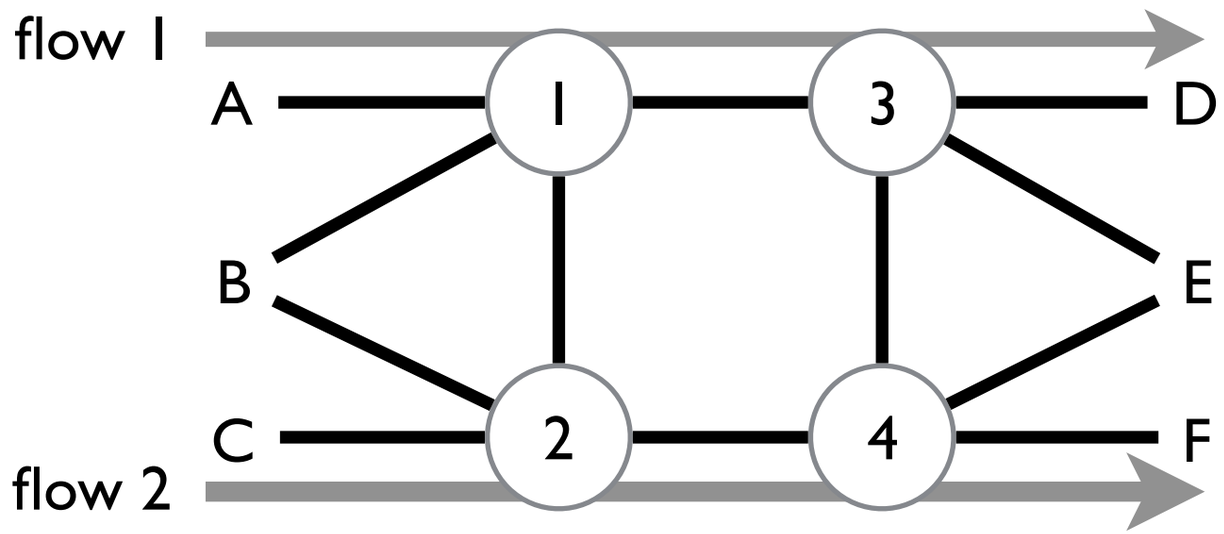
**cf**

configuration

| fid | sid | nid |
|---|---|---|
| 1 | 1 | 3 |
| 1 | 3 | D |

network event

control command



# abstraction: application view

firewall table

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

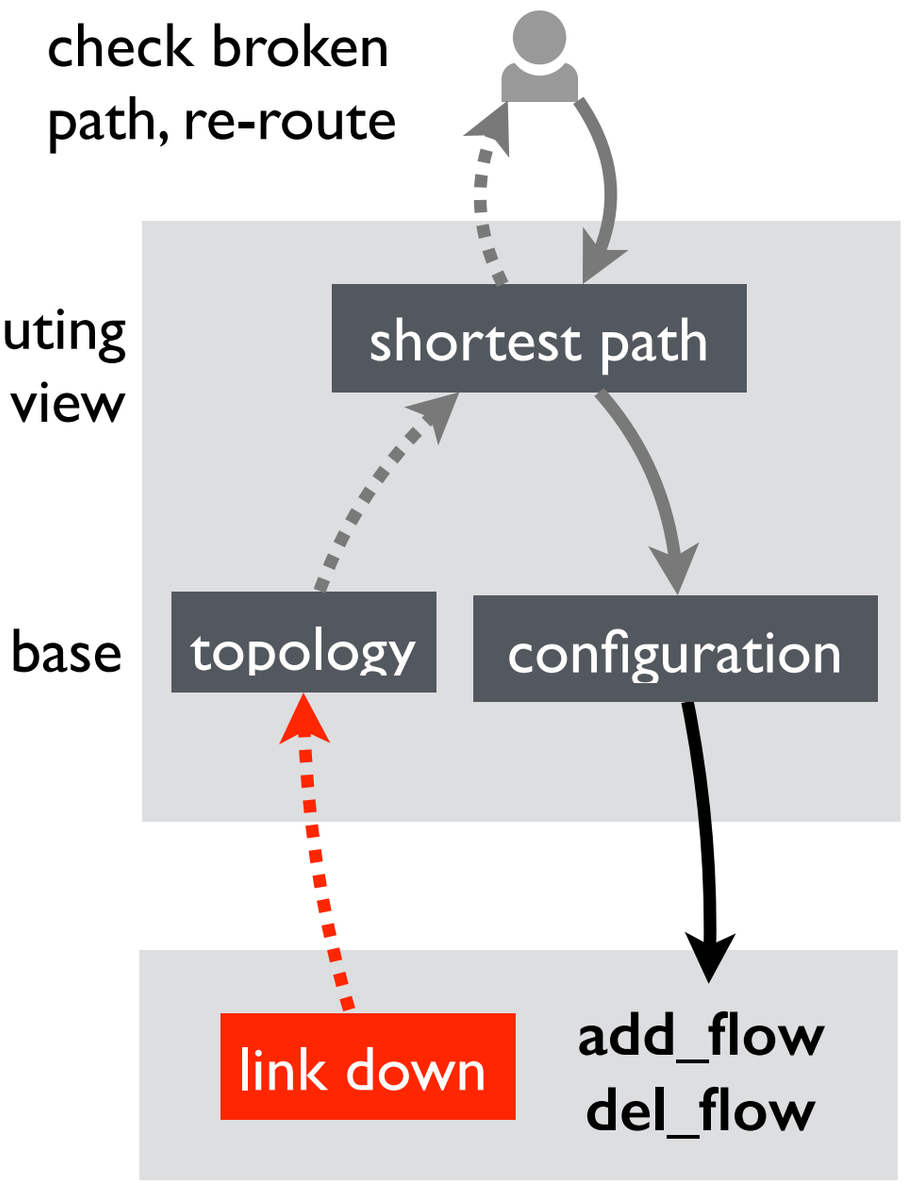
control loop: monitoring firewall view and repairing violation

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

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

- many more
  - routing, stateful firewall, service chain policy between subdomains …
- optimizing application by materializing views
  - (one order of magnitude) faster access with small maintenance overhead (.01~10ms)

# orchestration across representations




shortest path rule:
upon broken path, re-route

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

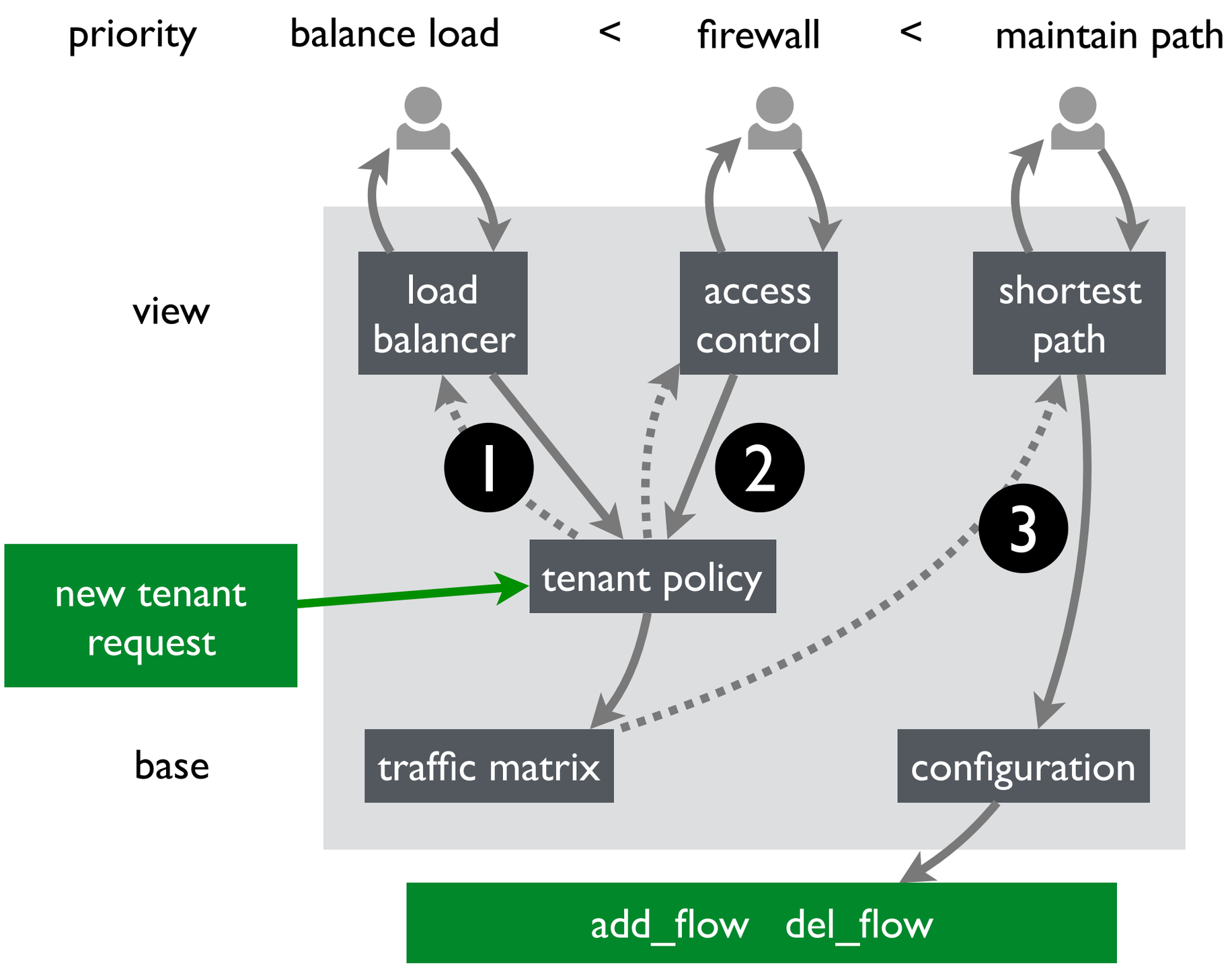
| | topology | | |
|---|---|---|---|
| | sid | nid | active |
| - | 172 | 39 | 1 |
| + | 172 | 39 | 0 |

| | 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=?)

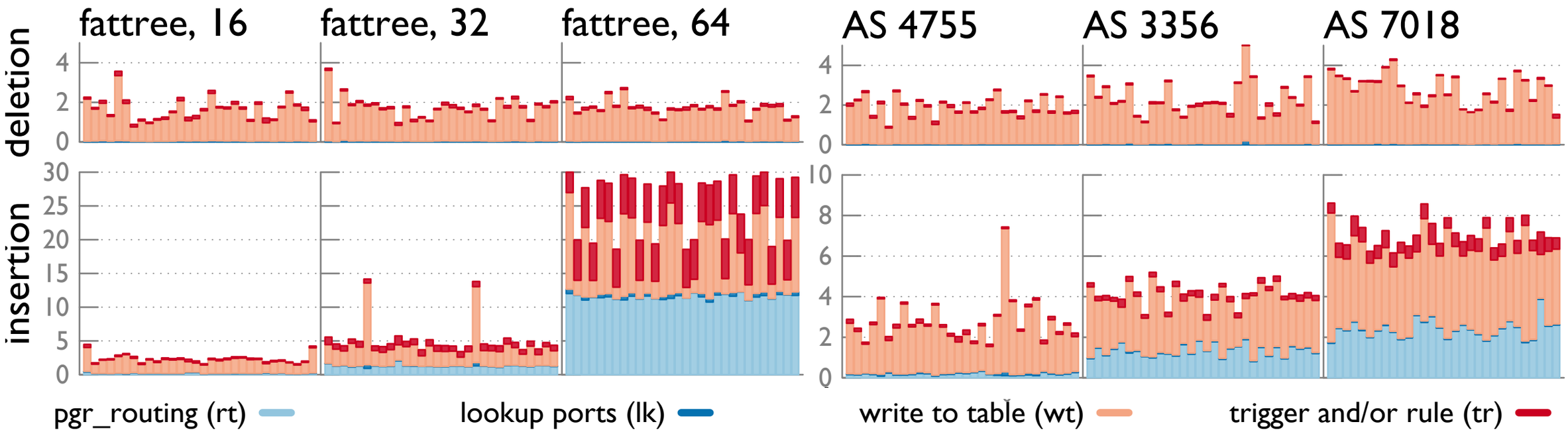
| | traffic matrix | | |
|---|---|---|---|
| | fid | sid | nid |
| + | … | 1238 | 1034 |

| | configuration | | |
|---|---|---|---|
| | fid | sid | nid |
| + | … | … | 1034 |

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

# evaluation

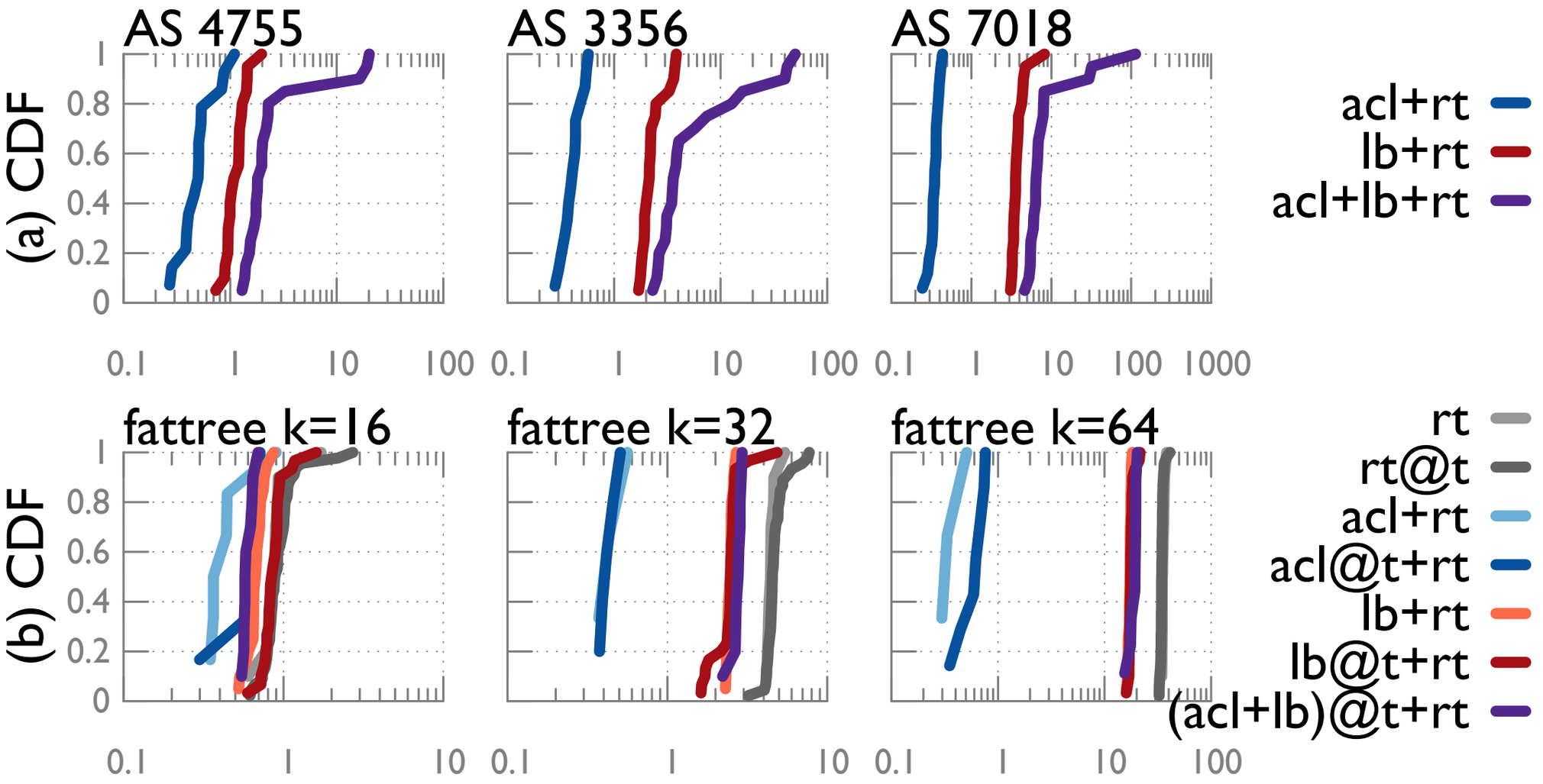
## profiling database delay — route insertion/deletion



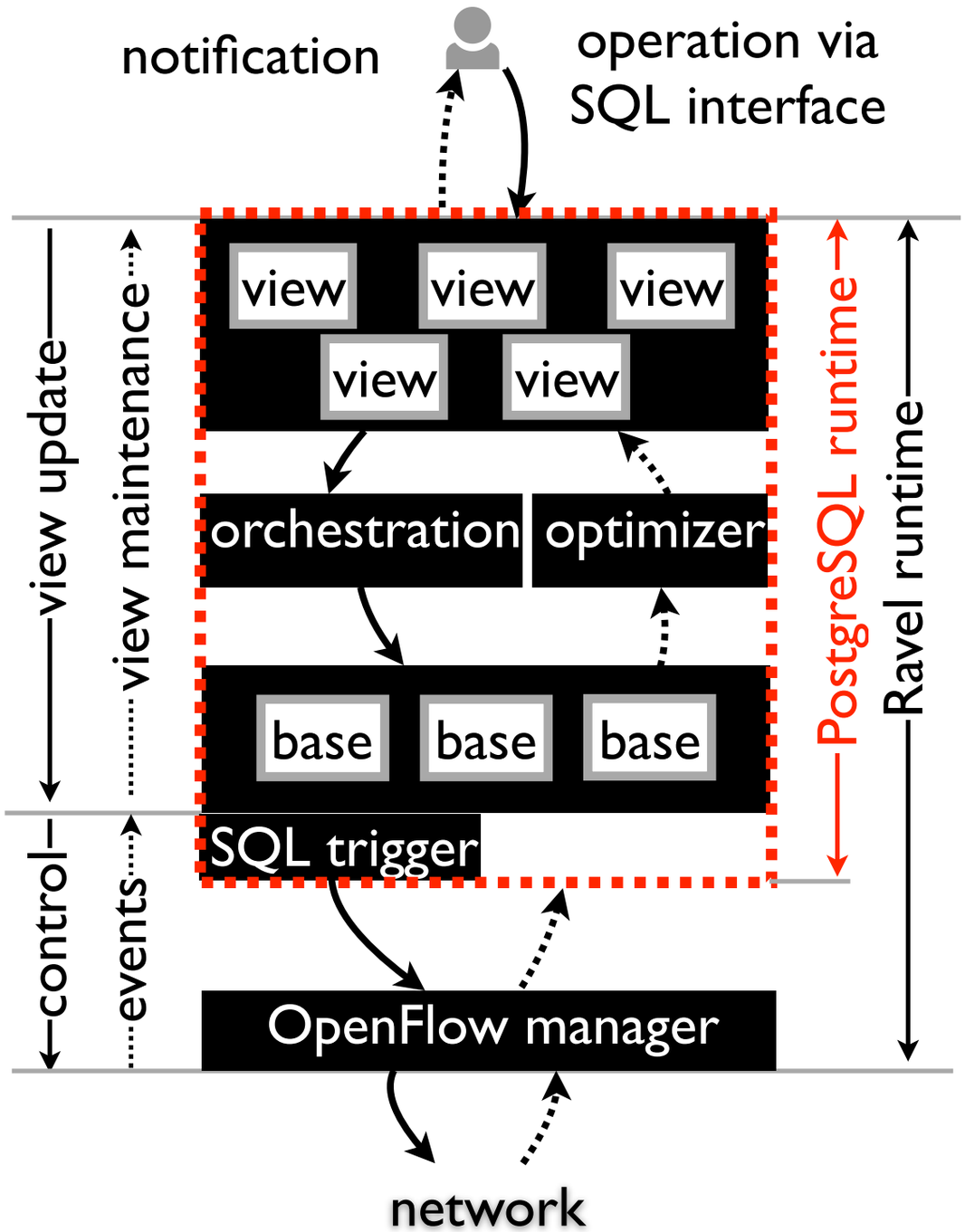

| fat-tree | | | ISP | | |
|---|---|---|---|---|---|
| k | switches | links | AS# | nodes | links |
| 16 | 320 | 3072 | 4755 | 142 | 258 |
| 32 | 1280 | 24576 | 3356 | 1772 | 13640 |
| 64 | 5120 | 196608 | 7018 | 25382 | 11292 |
| | | | 2914 | 5939 | 16520 |

# evaluation

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



# conclusion



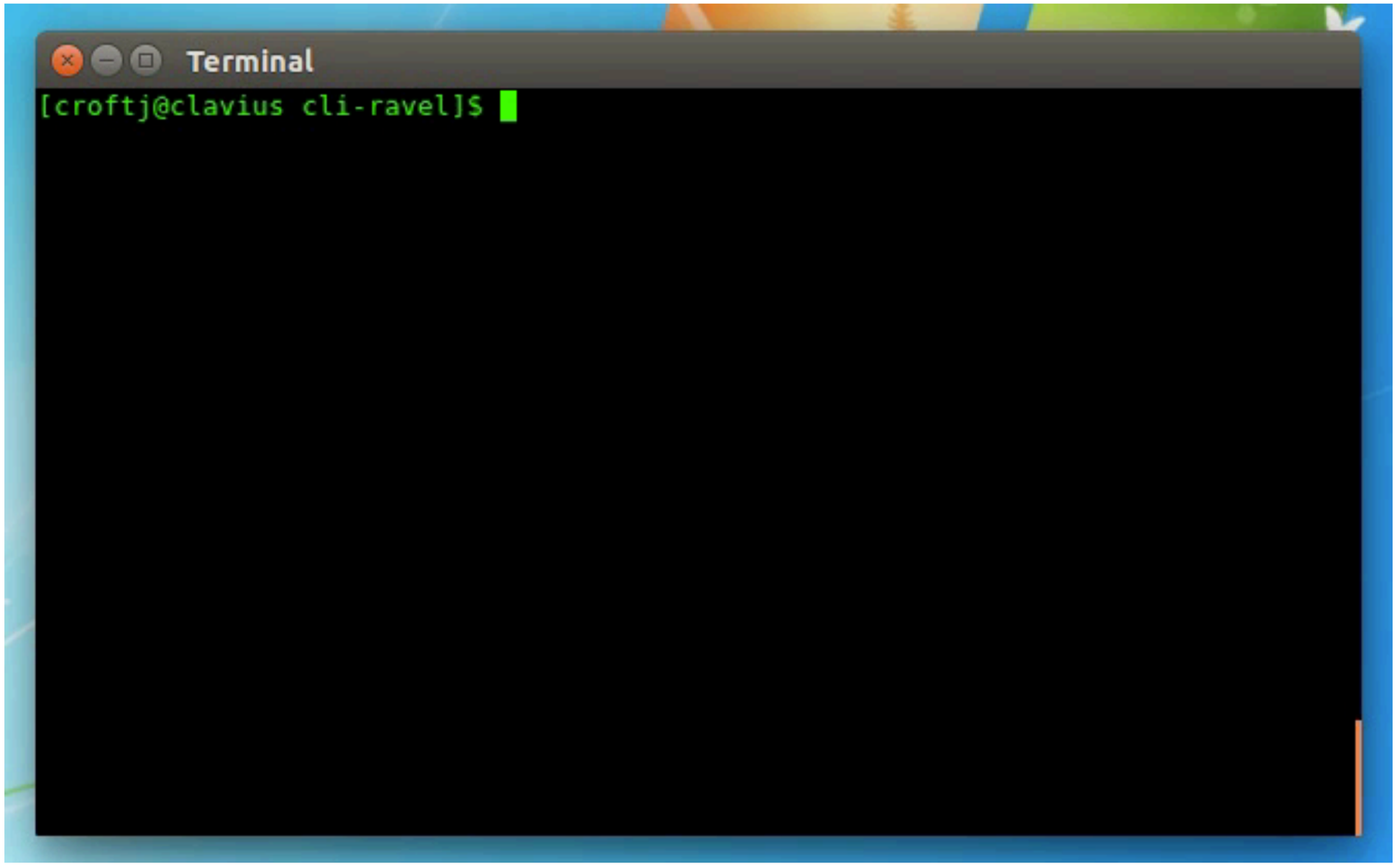

## flexible abstraction via SQL

- ad-hoc, user-defined, orchestratable
- promising performance

## looking forward

- application of database features
  - network-wide transaction
  - bootstrapping legacy networks
- enhancing database
  - better runtime: synthesizing orchestration
  - better control decision: property analysis
- interpretability
  - integrate foreign applications, plug-n-play 3rd party solvers

# demo



# playtime

download *Ravel* and install

ravel-net.org/download

start playing: tutorials, add your own app

ravel-net.org

more to explore

github.com/ravel-net