5590, fall 2016
software defined networking

anduo wang, Temple University

TTLMAN 402, R 17:30-20:00

some materials in this slide are based on lectures by

Jennifer Rexford https://www.cs.princeton.edu/courses/archive/fall13/cos597E/

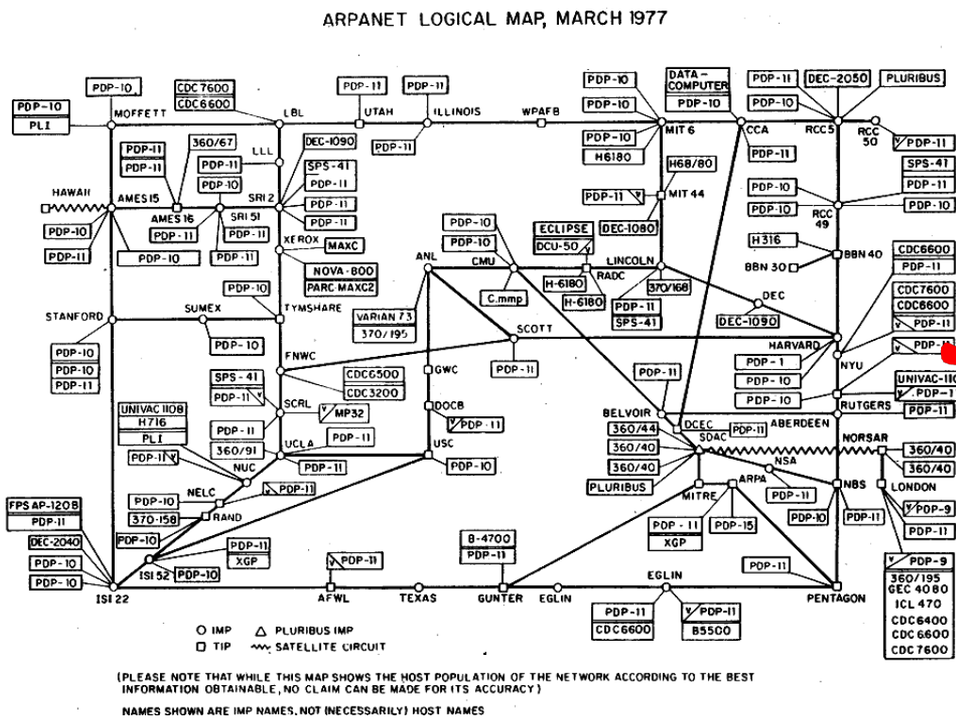
Nick Feamster http://noise.gatech.edu/classes/cs8803sdn/fall2014/

the state of networking

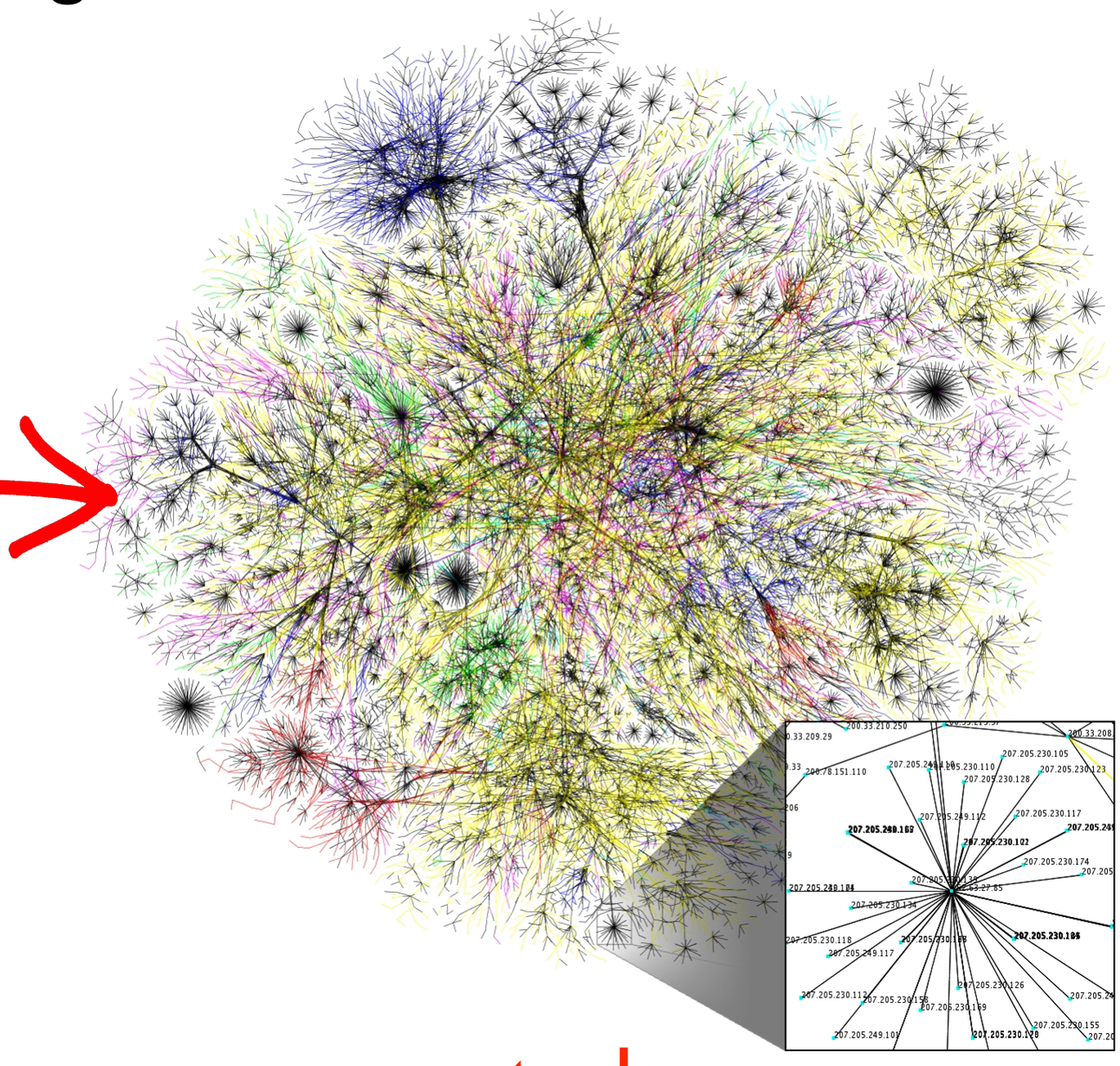
the Internet: a wonderful success

a remarkable story

- from research experiment to global infrastructure



ARPANET, 1977



today

source: https://en.wikipedia.org/wiki/Internet

the Internet: a wonderful success

innovations for everyday life

- Web, P2P, VoIP, social networking …

the Internet: a wonderful success

innovations for everyday life

- Web, P2P, VoIP, social networking …

the Internet: a wonderful success

innovations take rapid transitions

Ahmed Khurshid., et al. "VeriFlow: Verifying Network-Wide Invariants in Real Time"
source: https://www.usenix.org/conference/nsdi13/technical-sessions/presentation/khurshid
NSDI **2013**

3 years, $8.2 million

Veriflow Nabs $8.2 Million For Clever Ideas About Network Outage Prevention

JULY 19, 2016 BY DREW CONRY-MURRAY

Startup Veriflow Networks has landed $8.2 million in series A funding. The A round was led by Menlo Ventures, along with its existing investor New Enterprise Associates.

http://packetpushers.net/veriflow-nabs-8-2-million-clever-ideas-network-outage-prevention/

inside the 'Net': a different story

vendor lock-in

- specialized hardware
- protocols/software bundled with hardware
- slow innovation, deployment
- *$$$$$*

increasingly complex

- operators today are *masters of complexity*

discipline for networking?

operating systems
- time sharing

programming languages
- data abstractions

database management systems
- data independence

networking
- lack of discipline, but bags of protocols …

to do
- watch Scott Shenker's talk on "The Future of Networking, and the Past of Protocols" https://youtu.be/YHeyuD89n1Y

networking needs …

break vendor lock-in

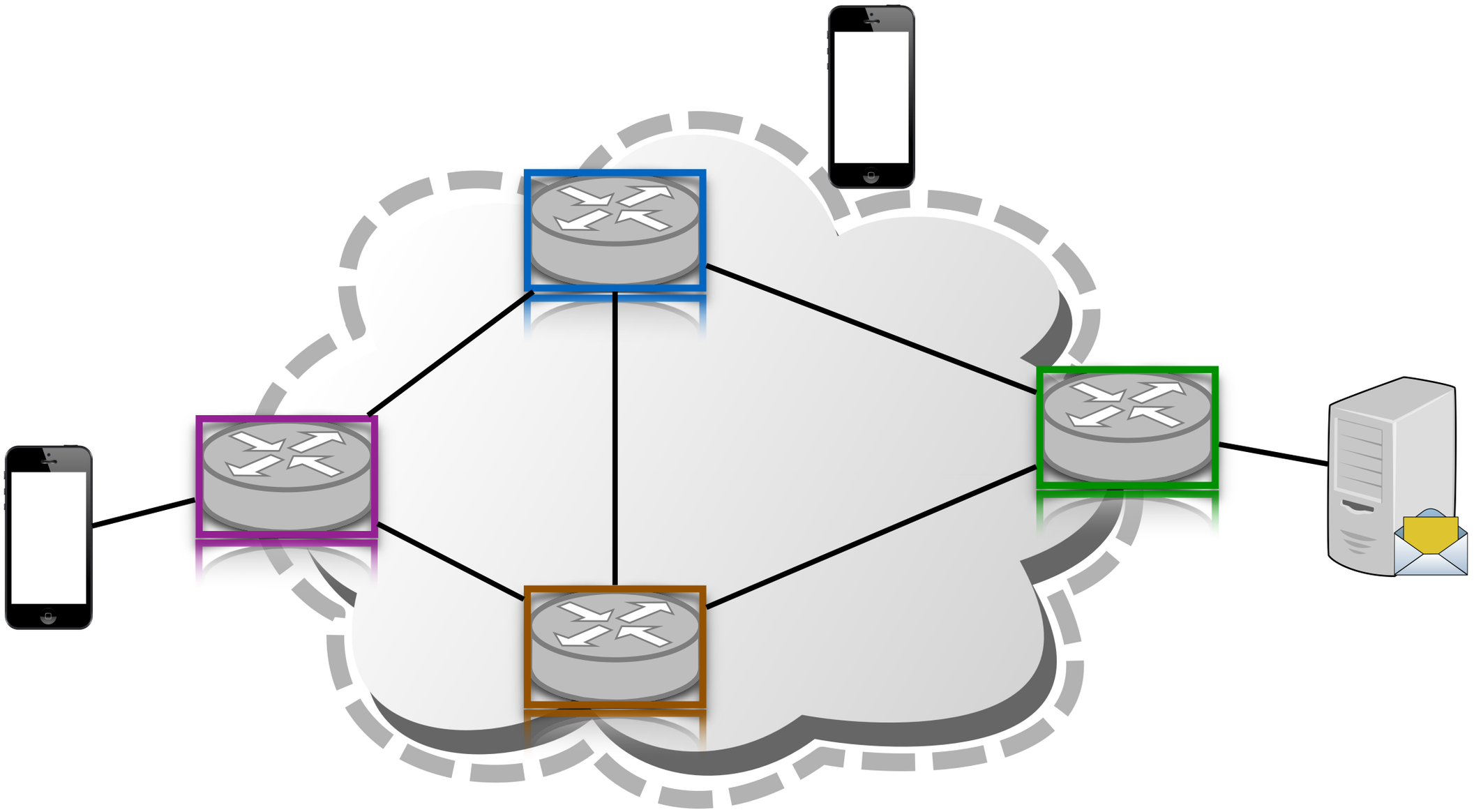
- freedom from suppliers
- freedom from low-level box by box configuration
- freedom of adding new services

introduce disciplines

- systematic principles that guide networking practice

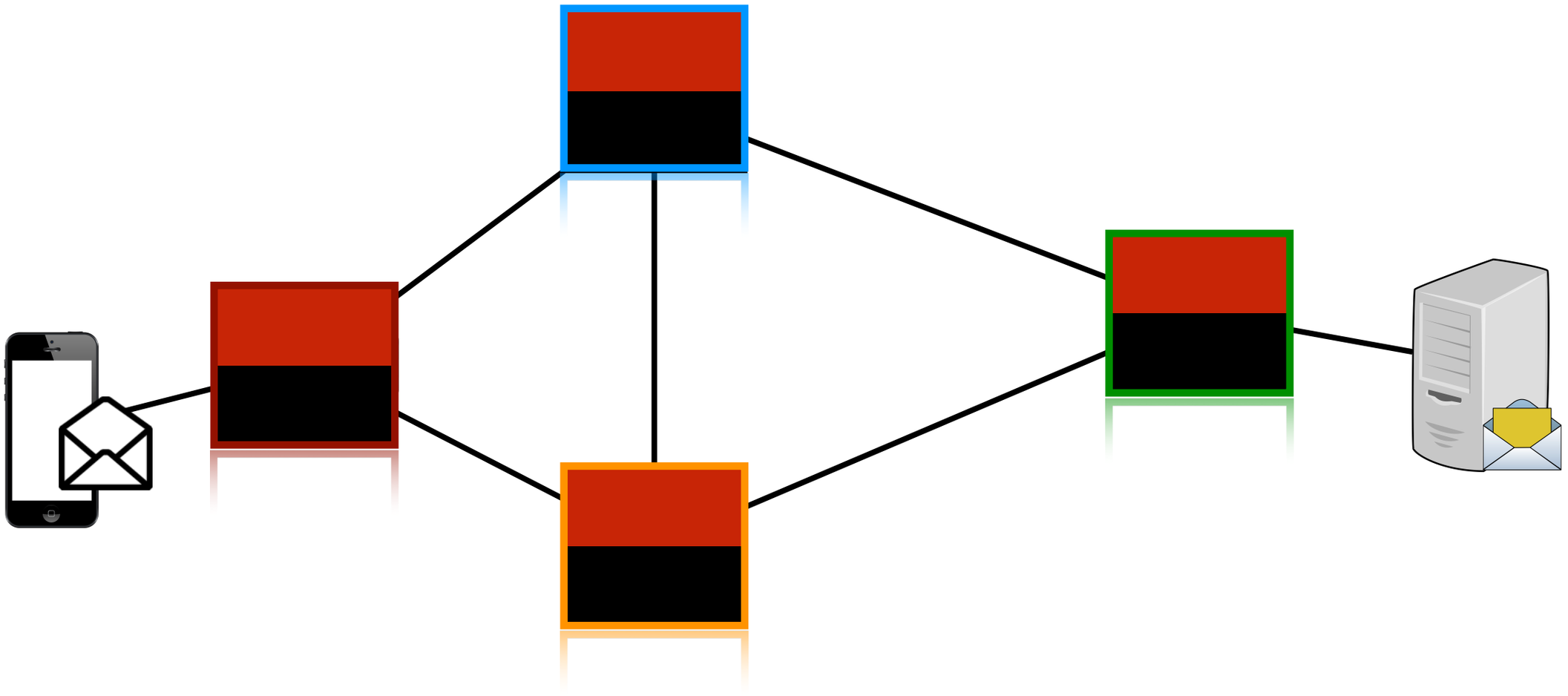
software defined networking (SDN)

software defined networks



software defined networks

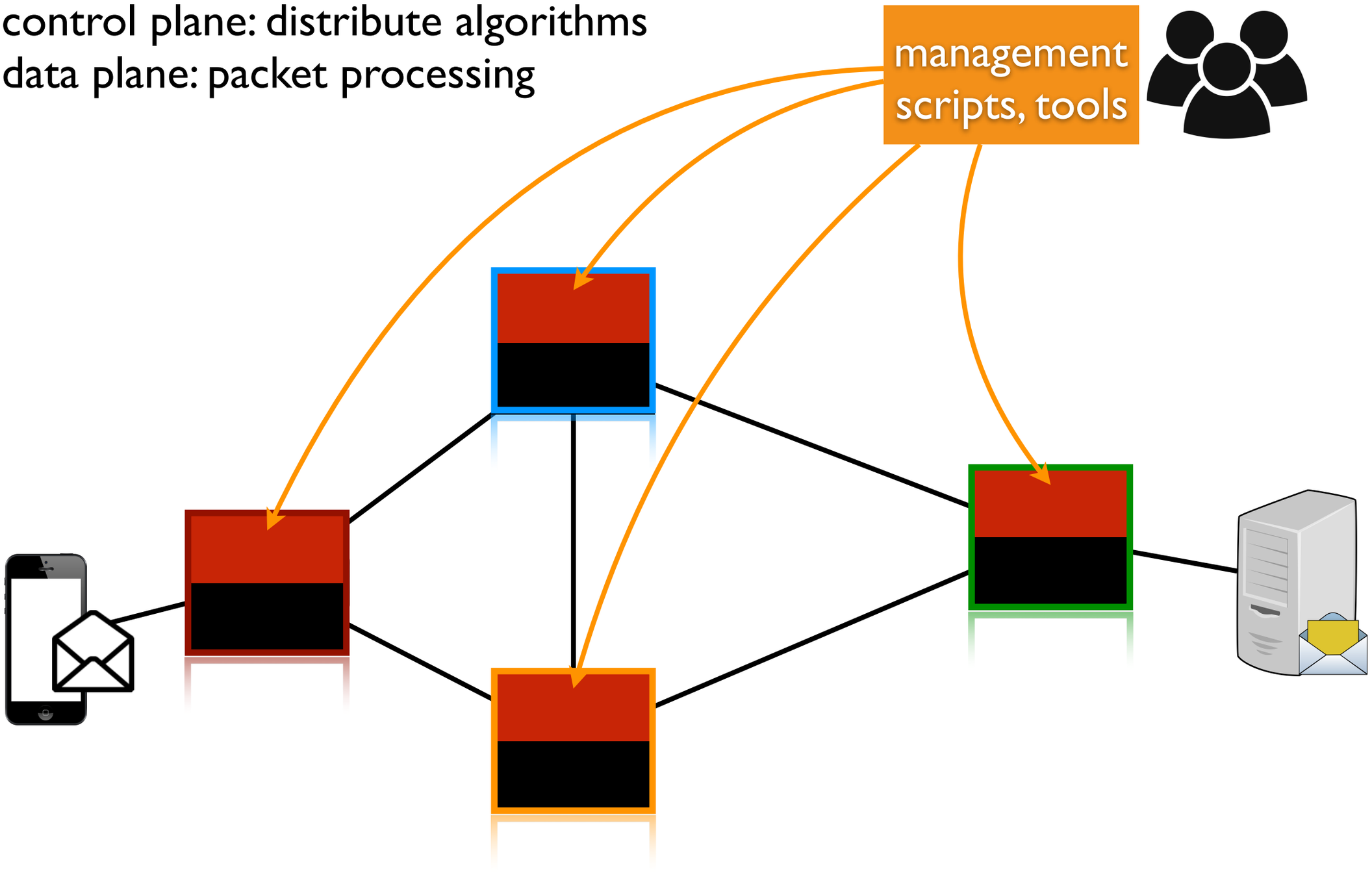
control plane: distribute algorithms
data plane: packet processing



software defined networks

control plane: distribute algorithms

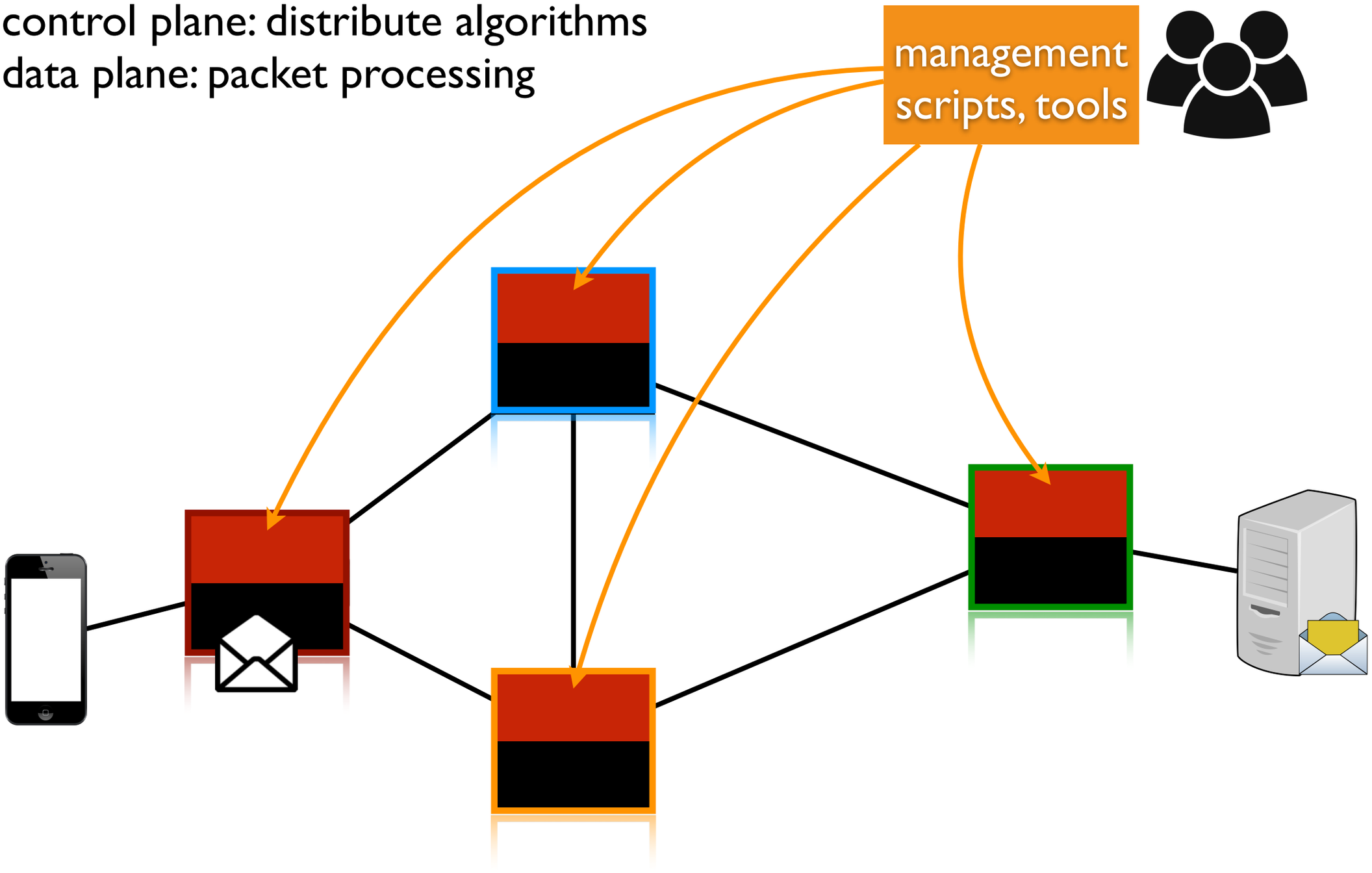
data plane: packet processing



software defined networks

control plane: distribute algorithms

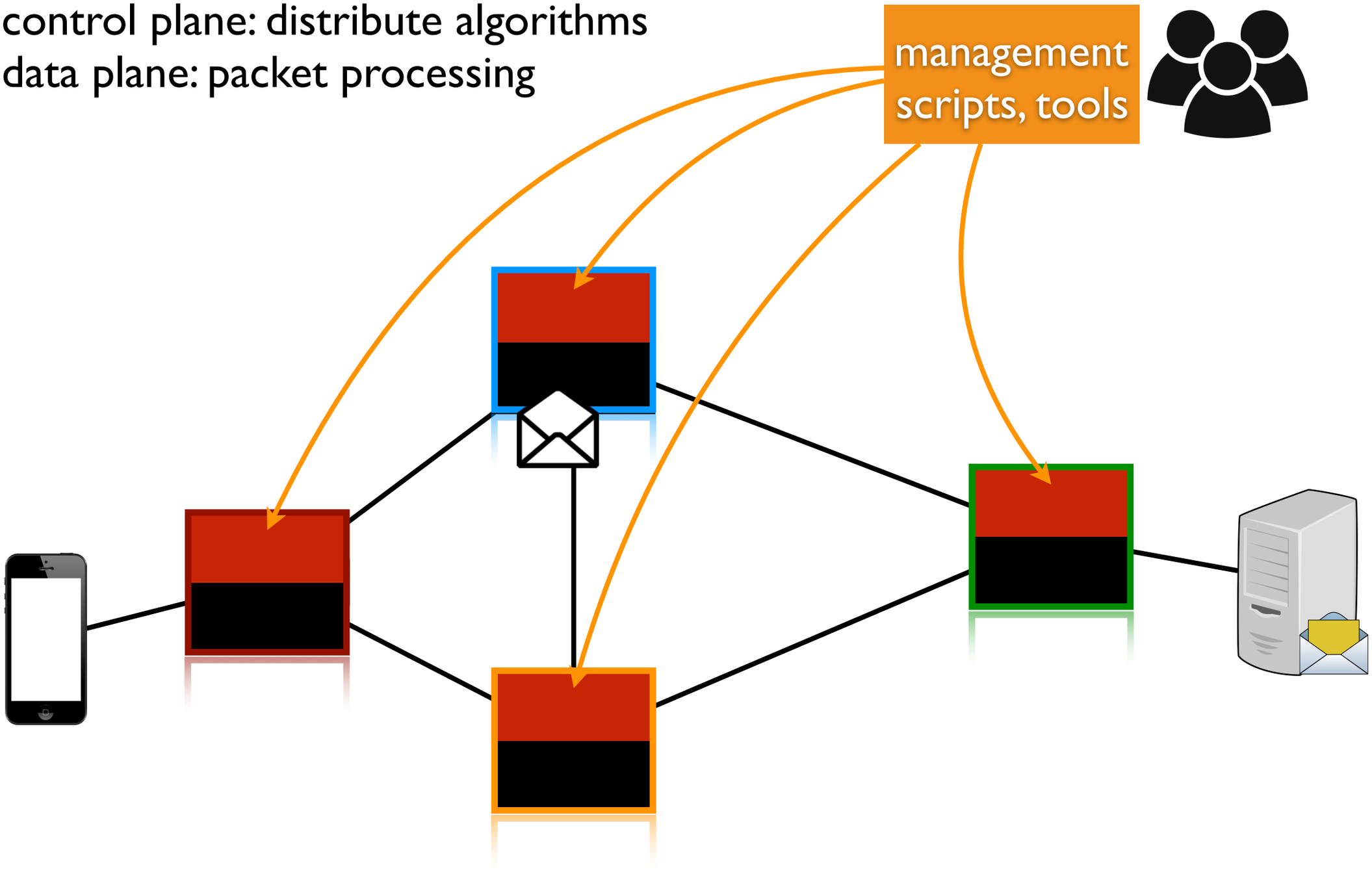
data plane: packet processing



software defined networks

control plane: distribute algorithms

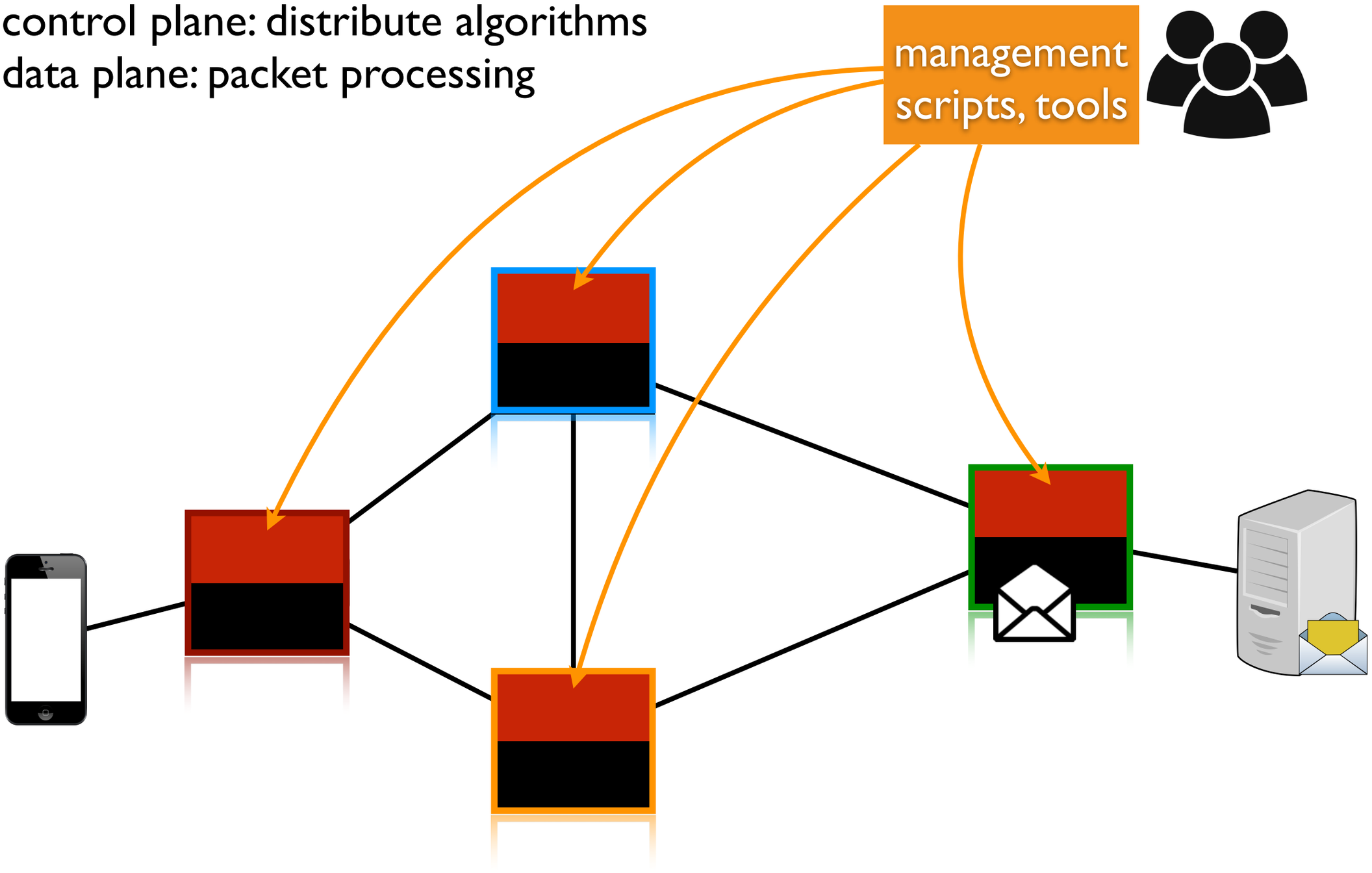
data plane: packet processing



software defined networks

control plane: distribute algorithms

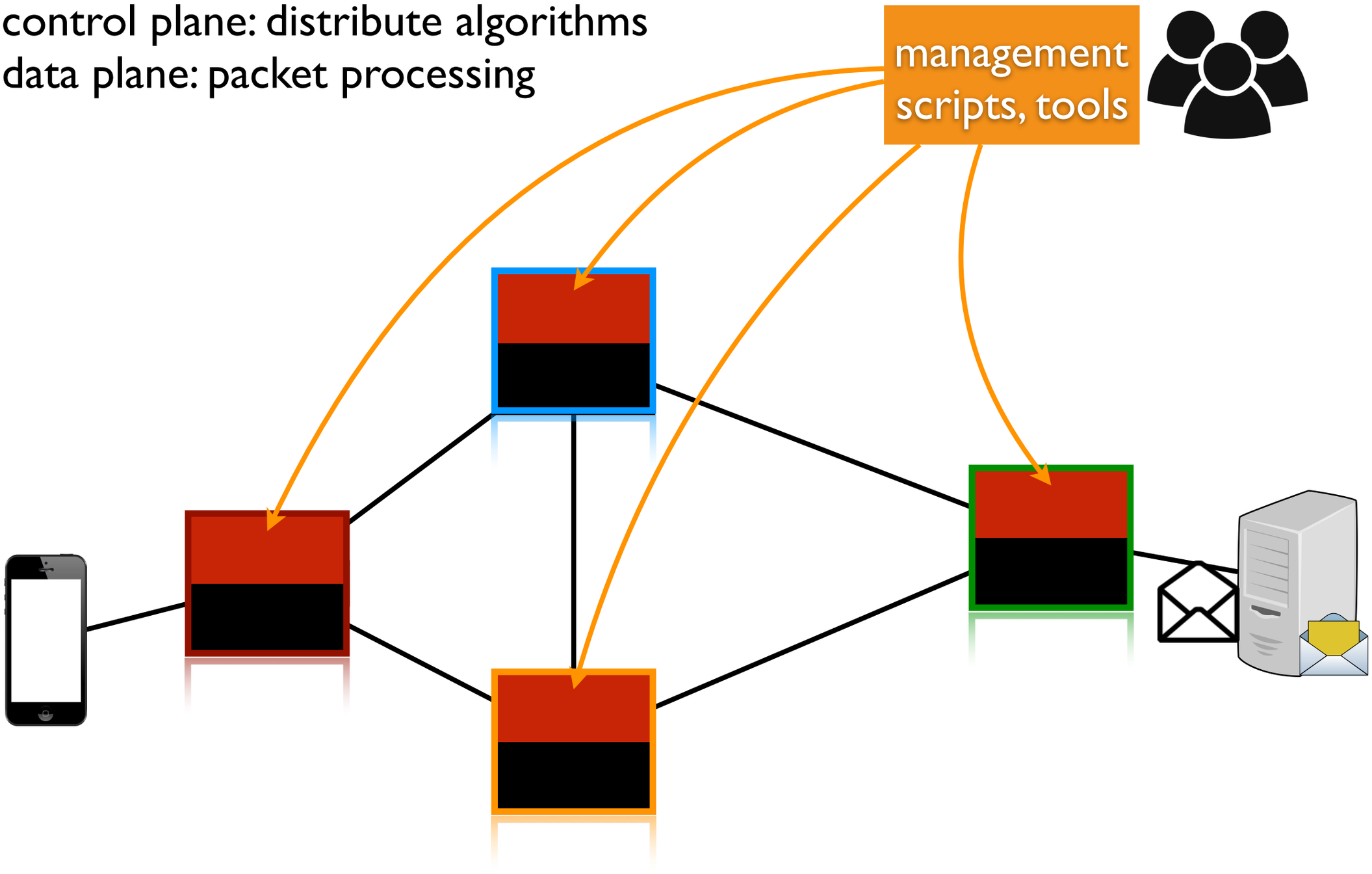
data plane: packet processing



software defined networks

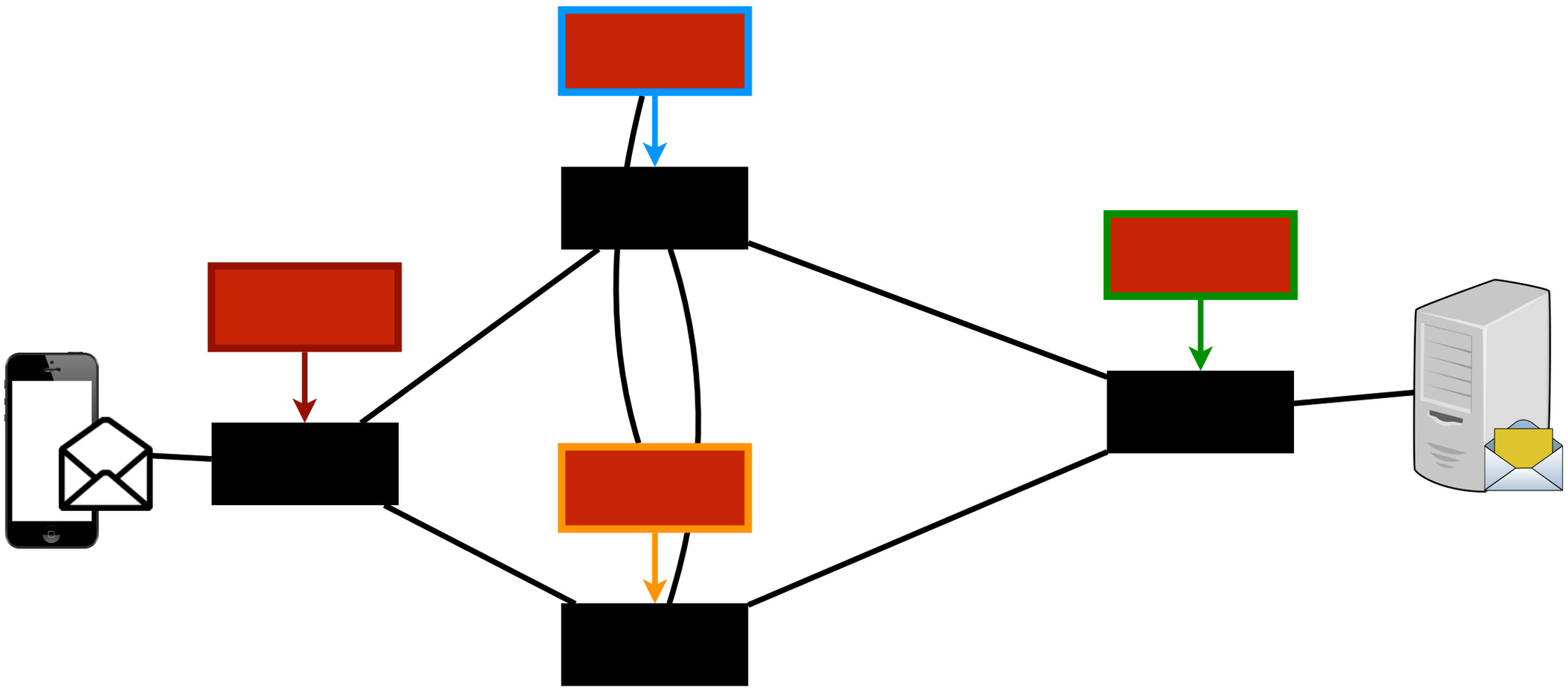
control plane: distribute algorithms

data plane: packet processing



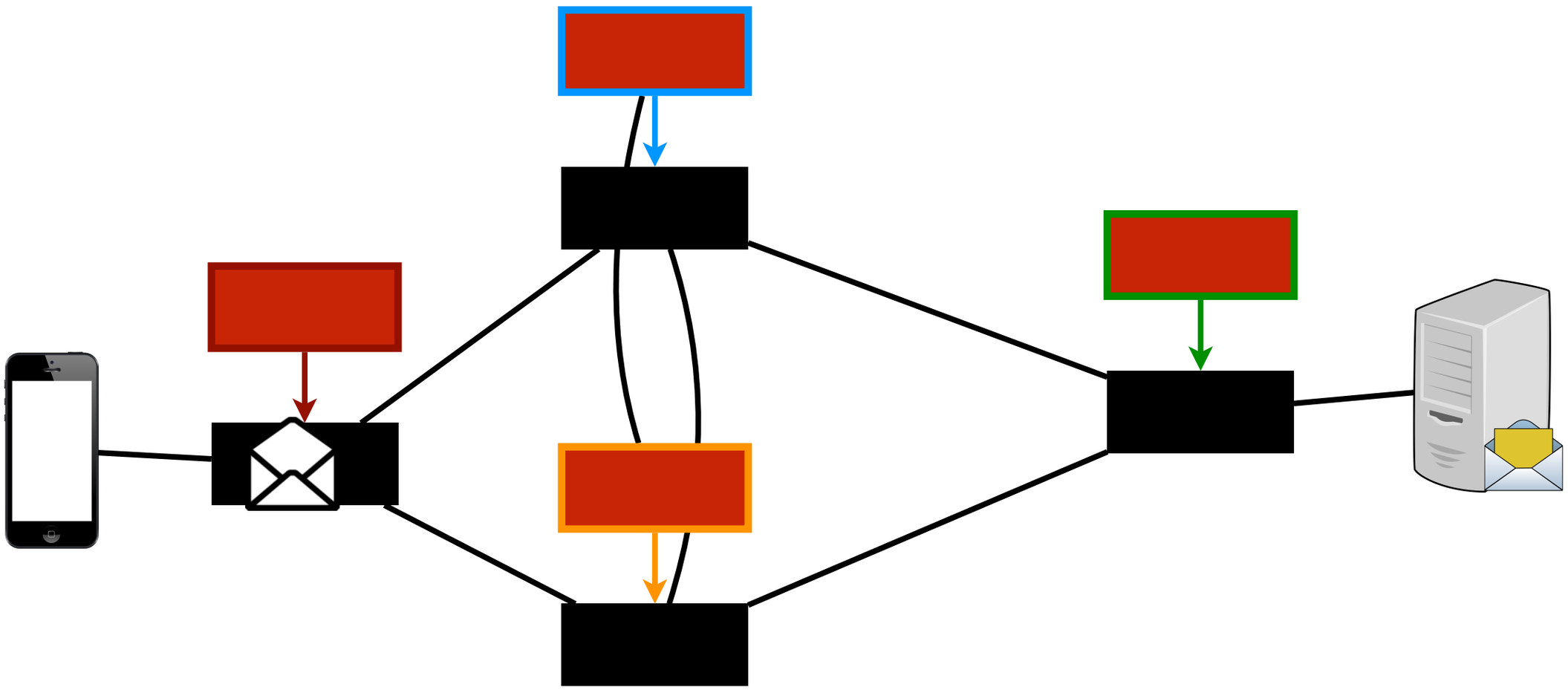
software defined networks

decouple control and data planes



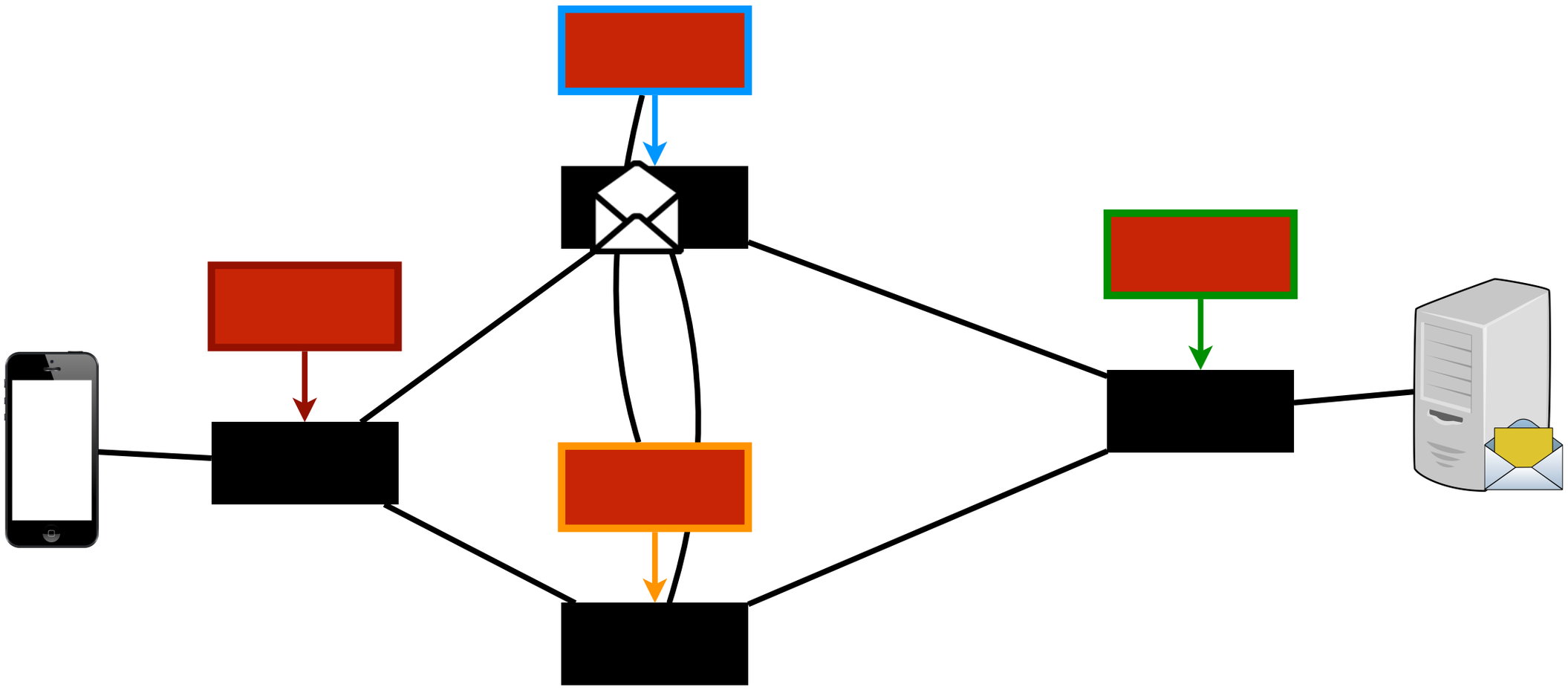
software defined networks

decouple control and data planes



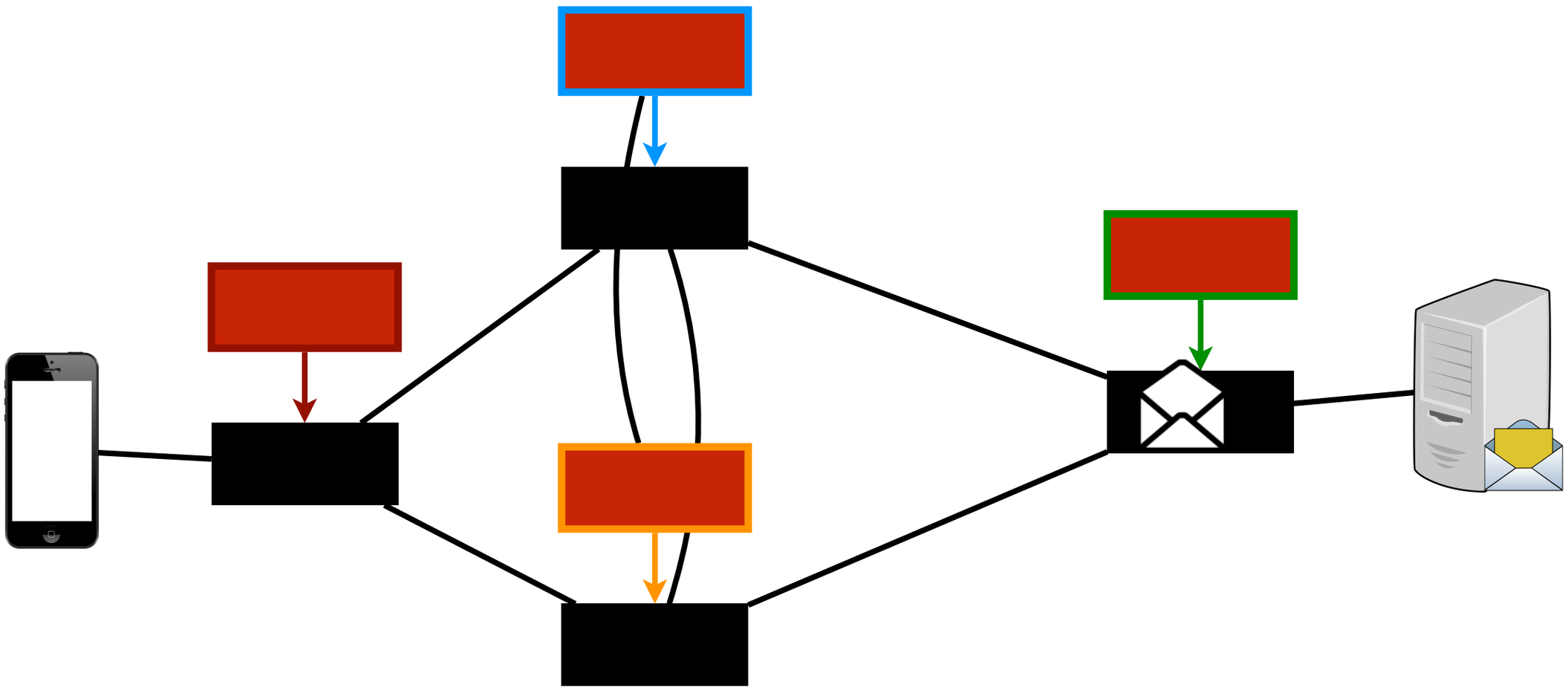
software defined networks

decouple control and data planes



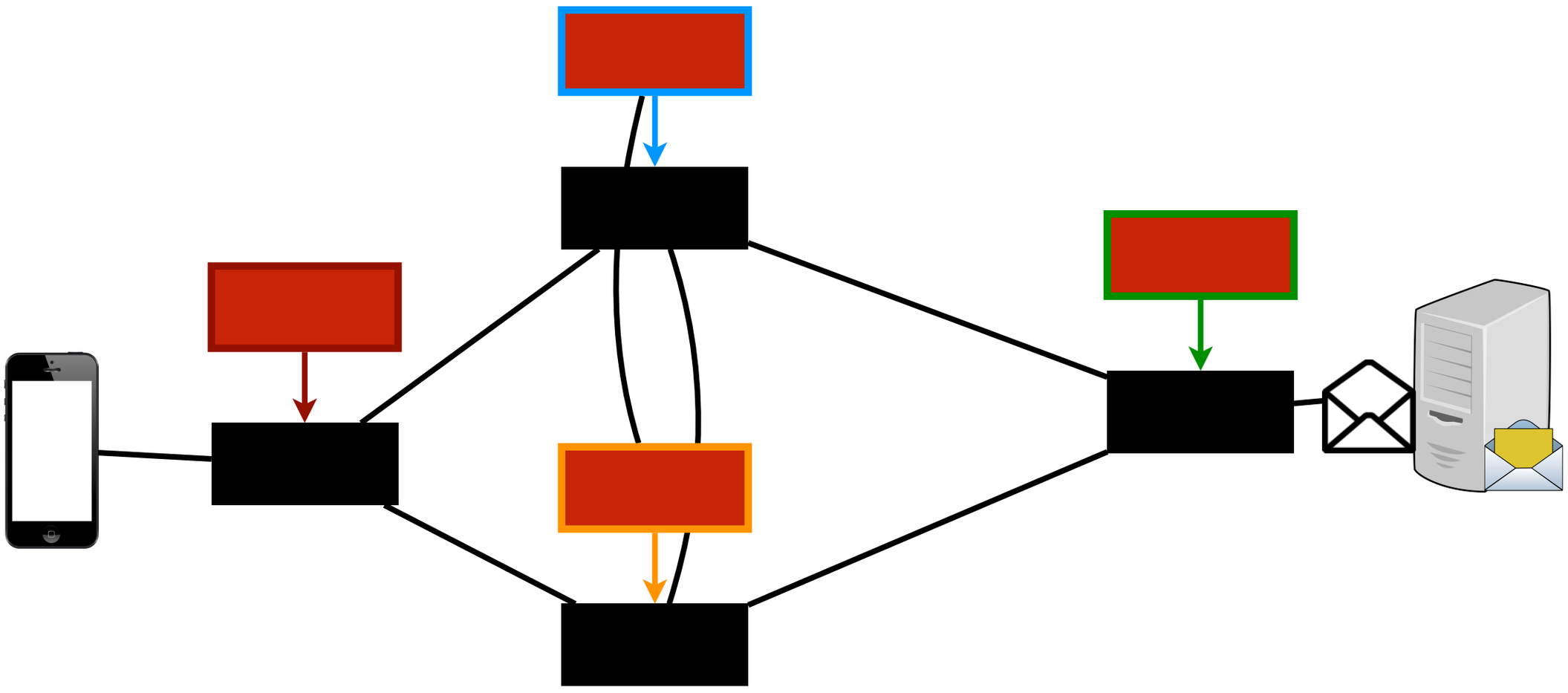
software defined networks

decouple control and data planes



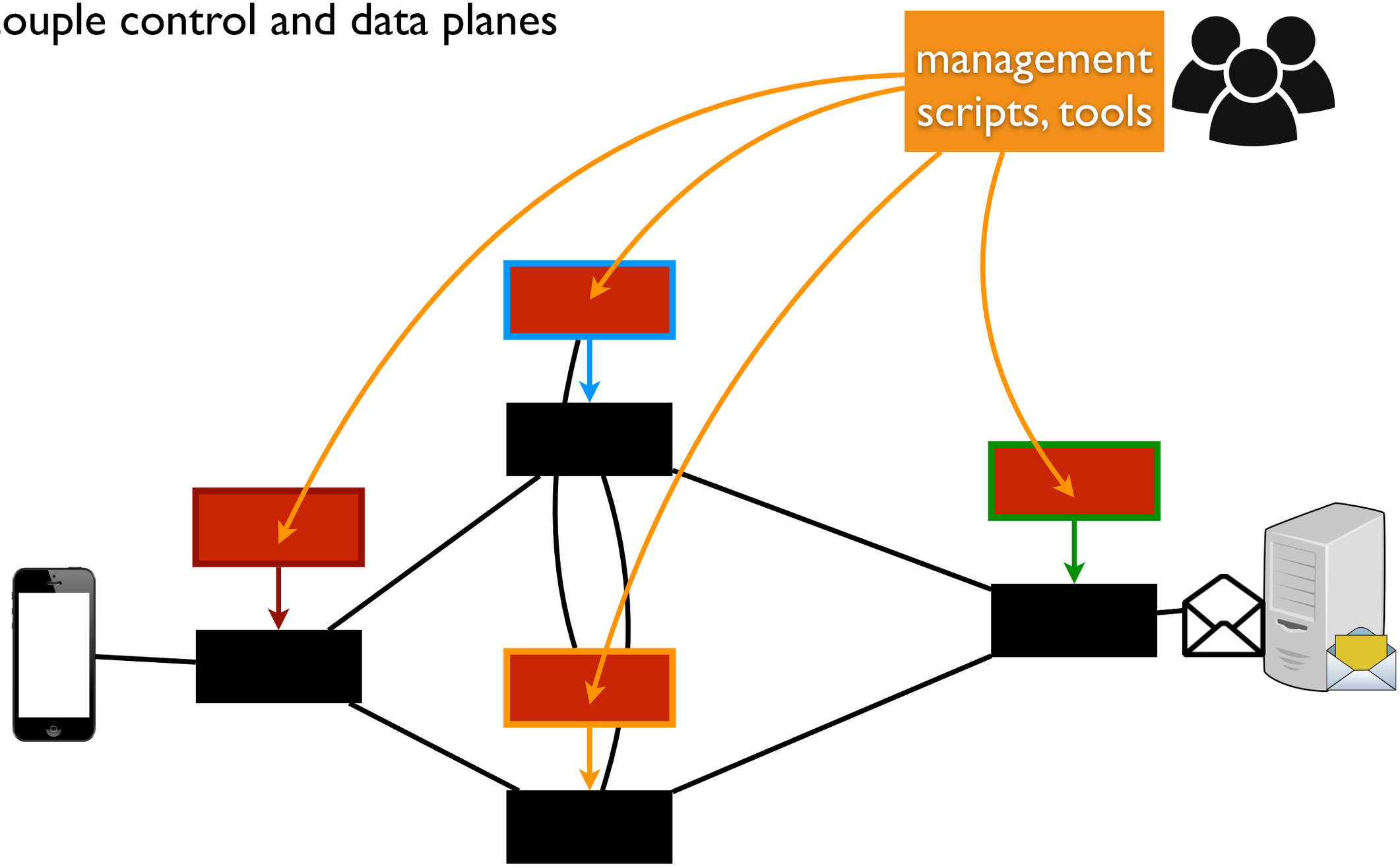
software defined networks

decouple control and data planes



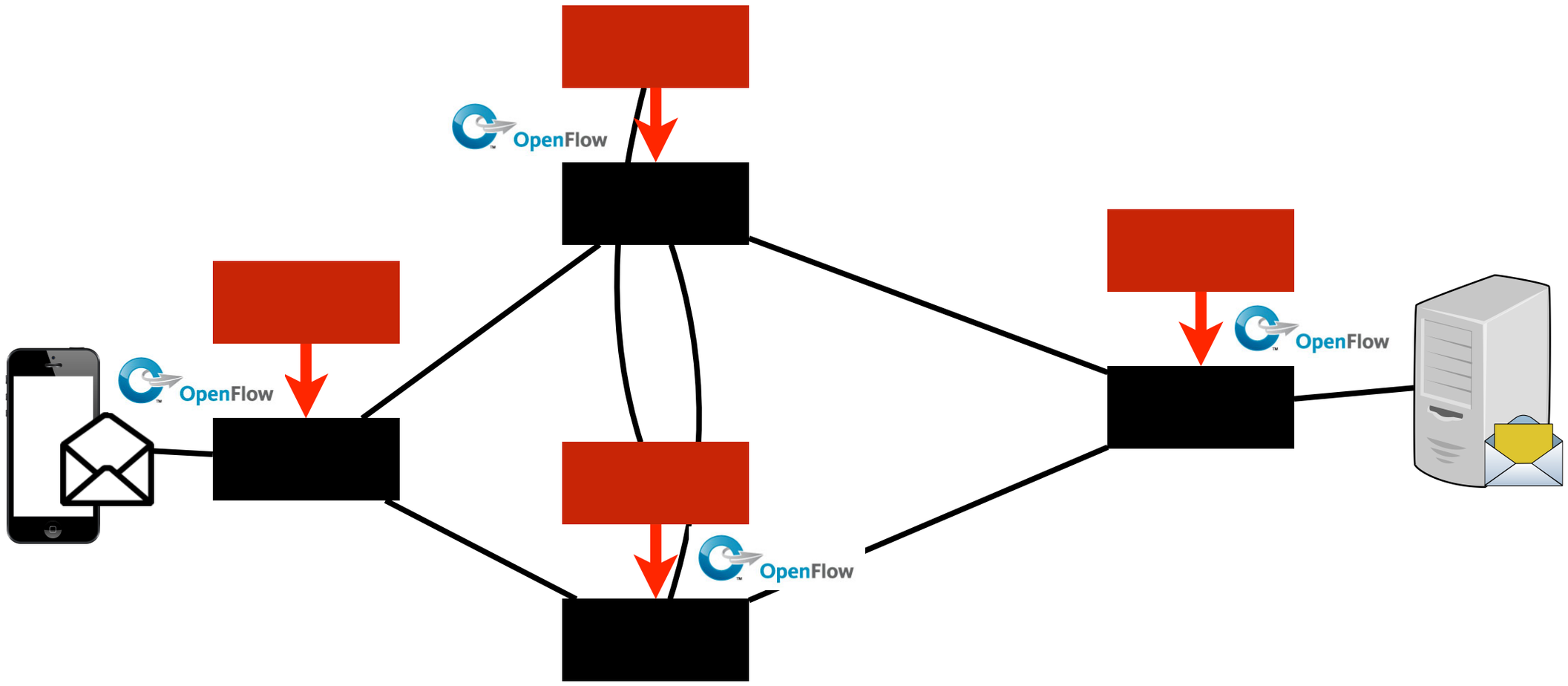
software defined networks

decouple control and data planes



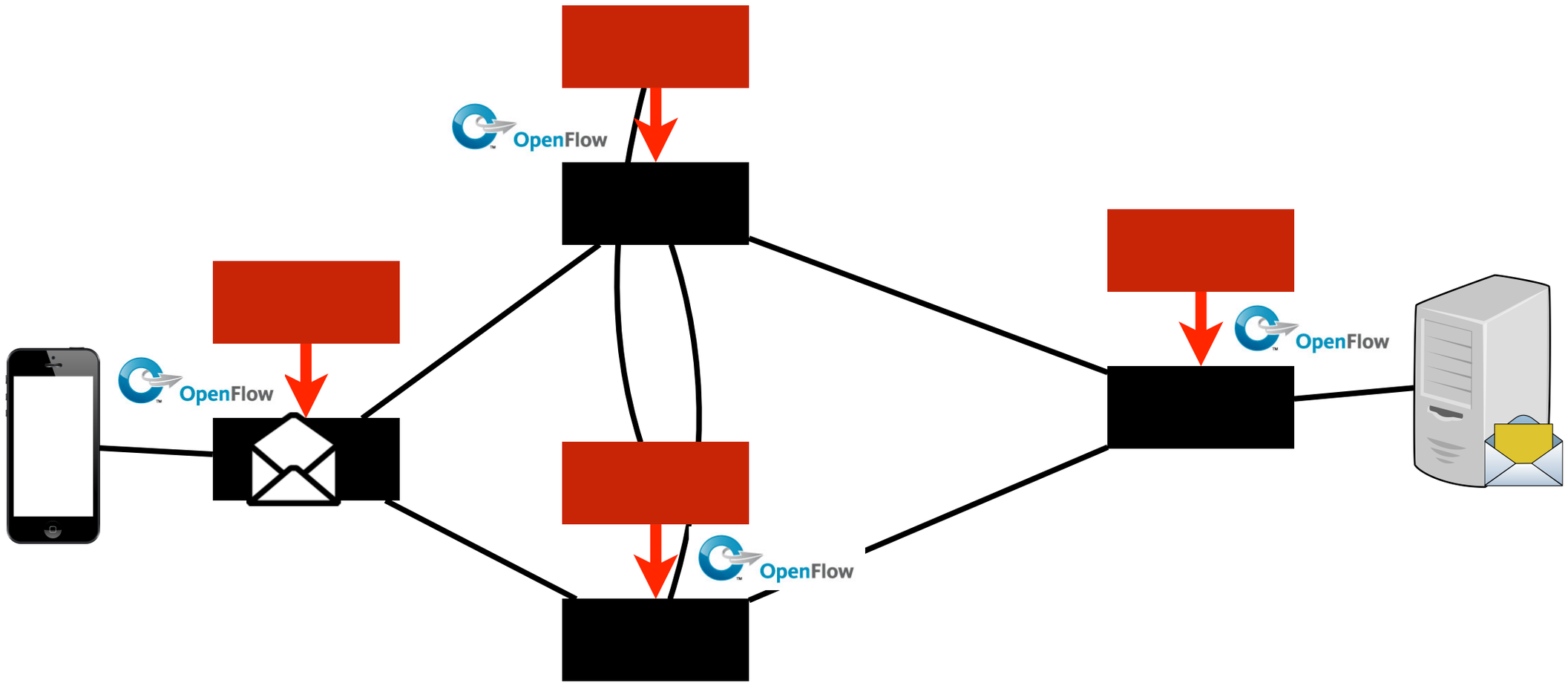
open dataplane API

decouple control and data planes
by providing **open standard API**
OpenFlow



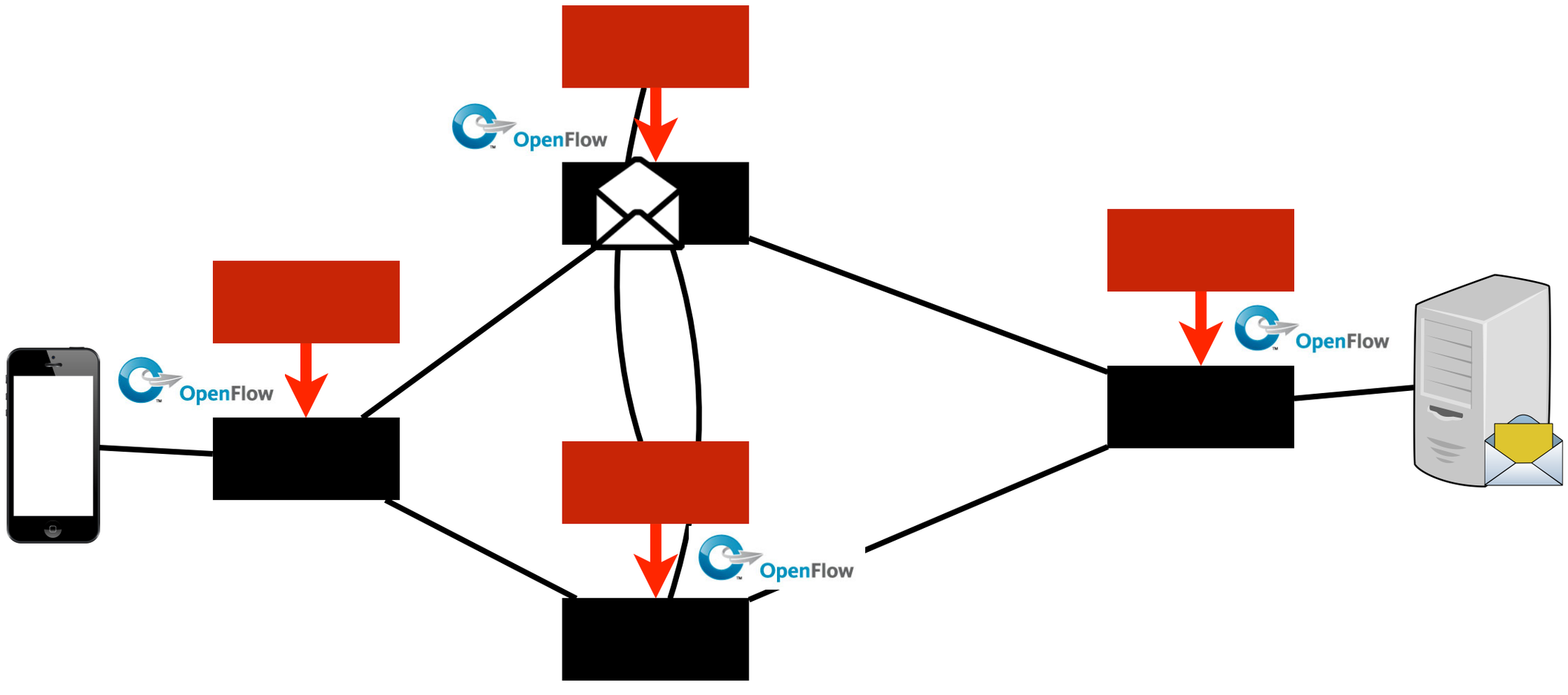
open dataplane API

decouple control and data planes
by providing **open standard API**
OpenFlow



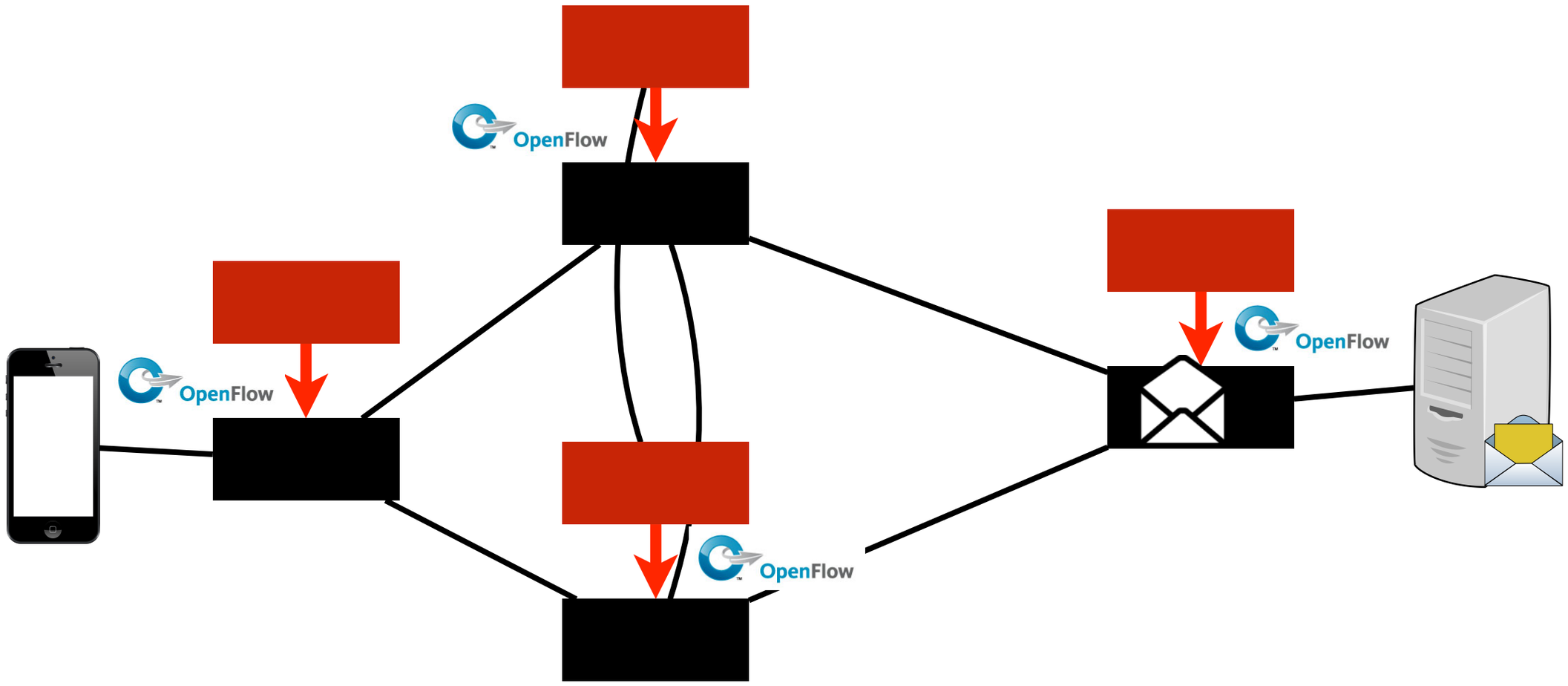
open dataplane API

decouple control and data planes
by providing **open standard API**
OpenFlow



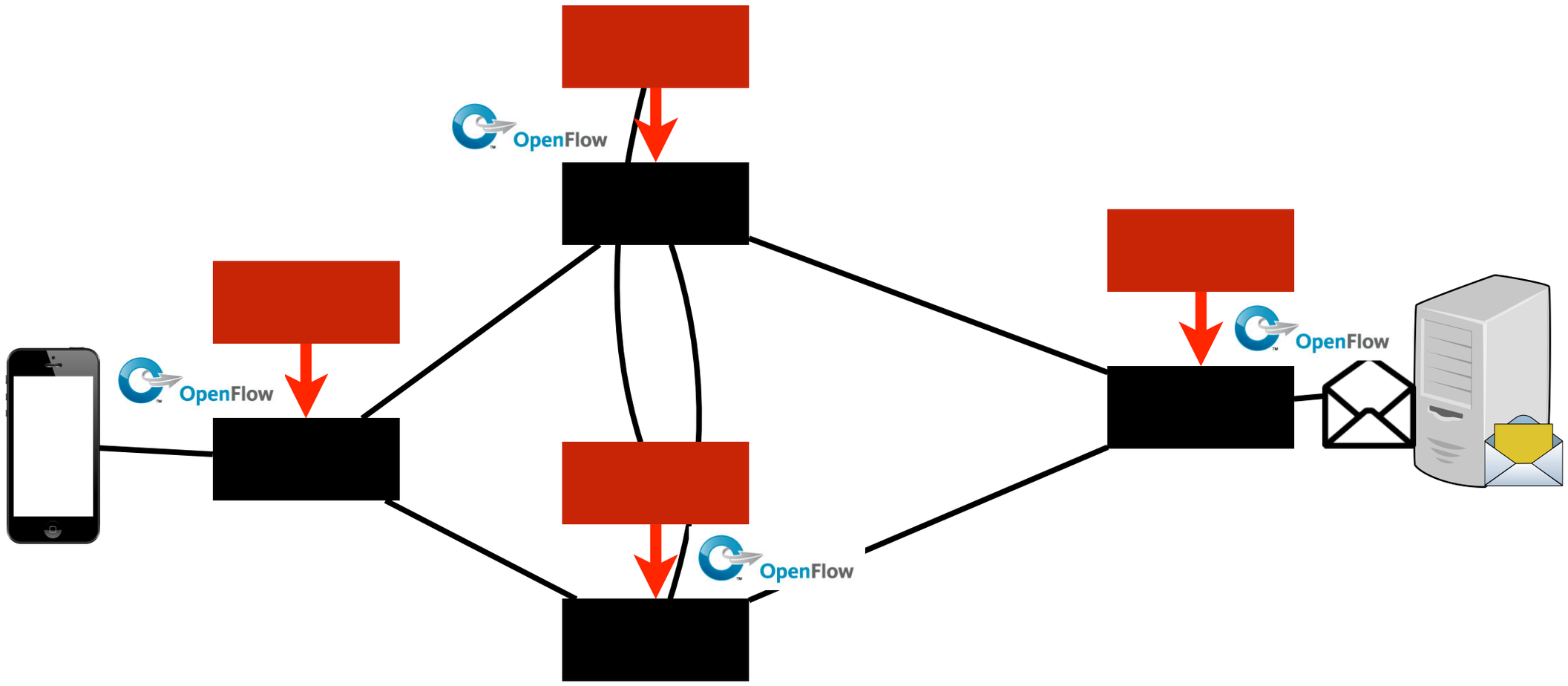
open dataplane API

decouple control and data planes
by providing **open standard API**
OpenFlow



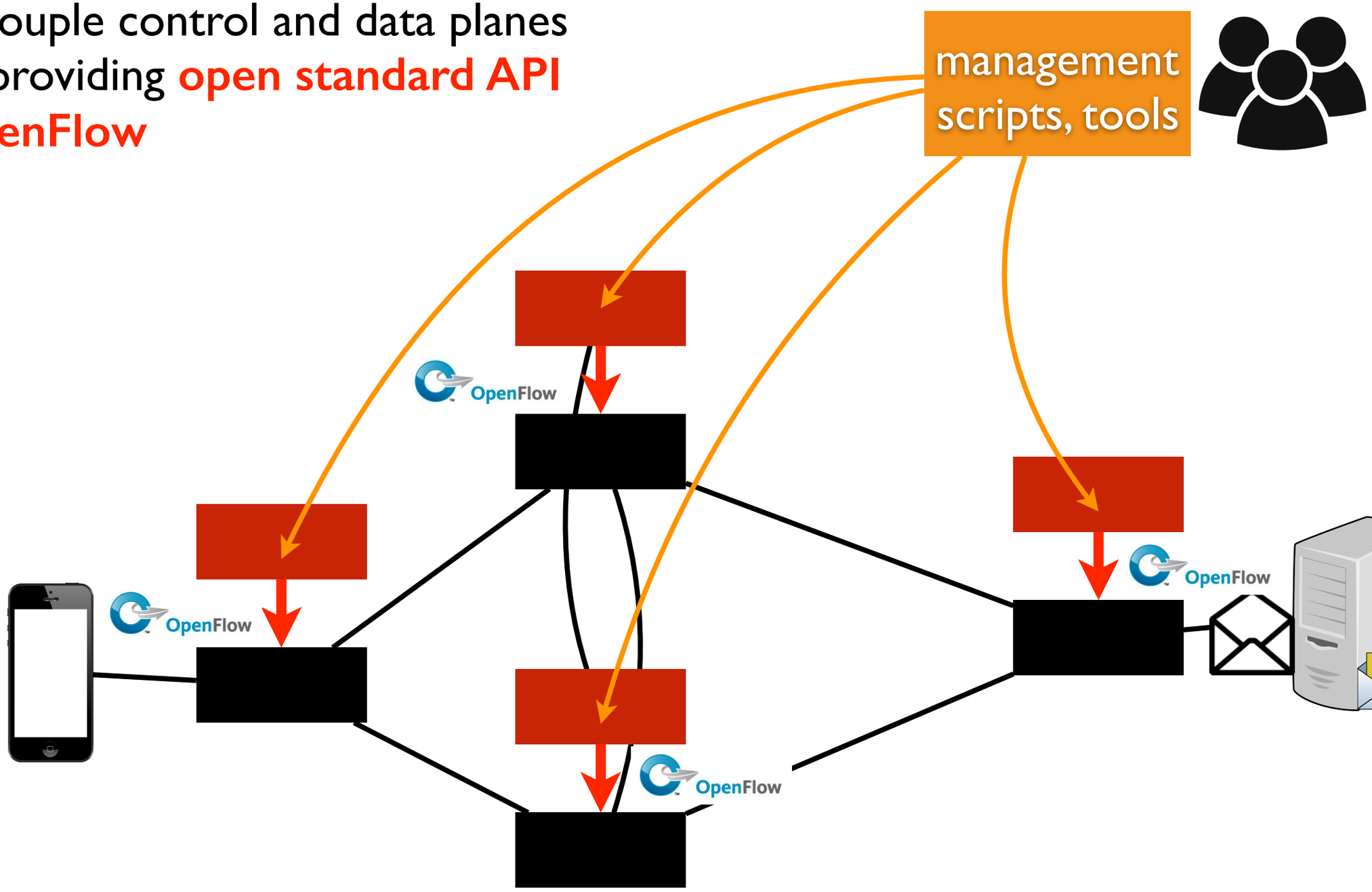
open dataplane API

decouple control and data planes
by providing **open standard API**
OpenFlow



open dataplane API

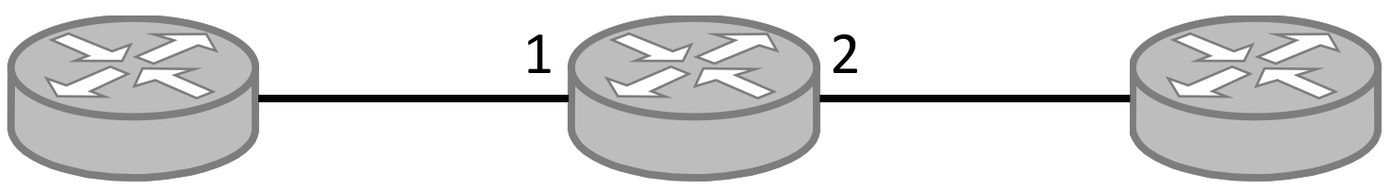
decouple control and data planes
by providing **open standard API**
OpenFlow



OpenFlow: simple open dataplane API

prioritized list of rules

- pattern → action
  - pattern: match packet header bits
  - actions: drop, forward, modify, send to controller
- priority: disambiguate overlapping patterns

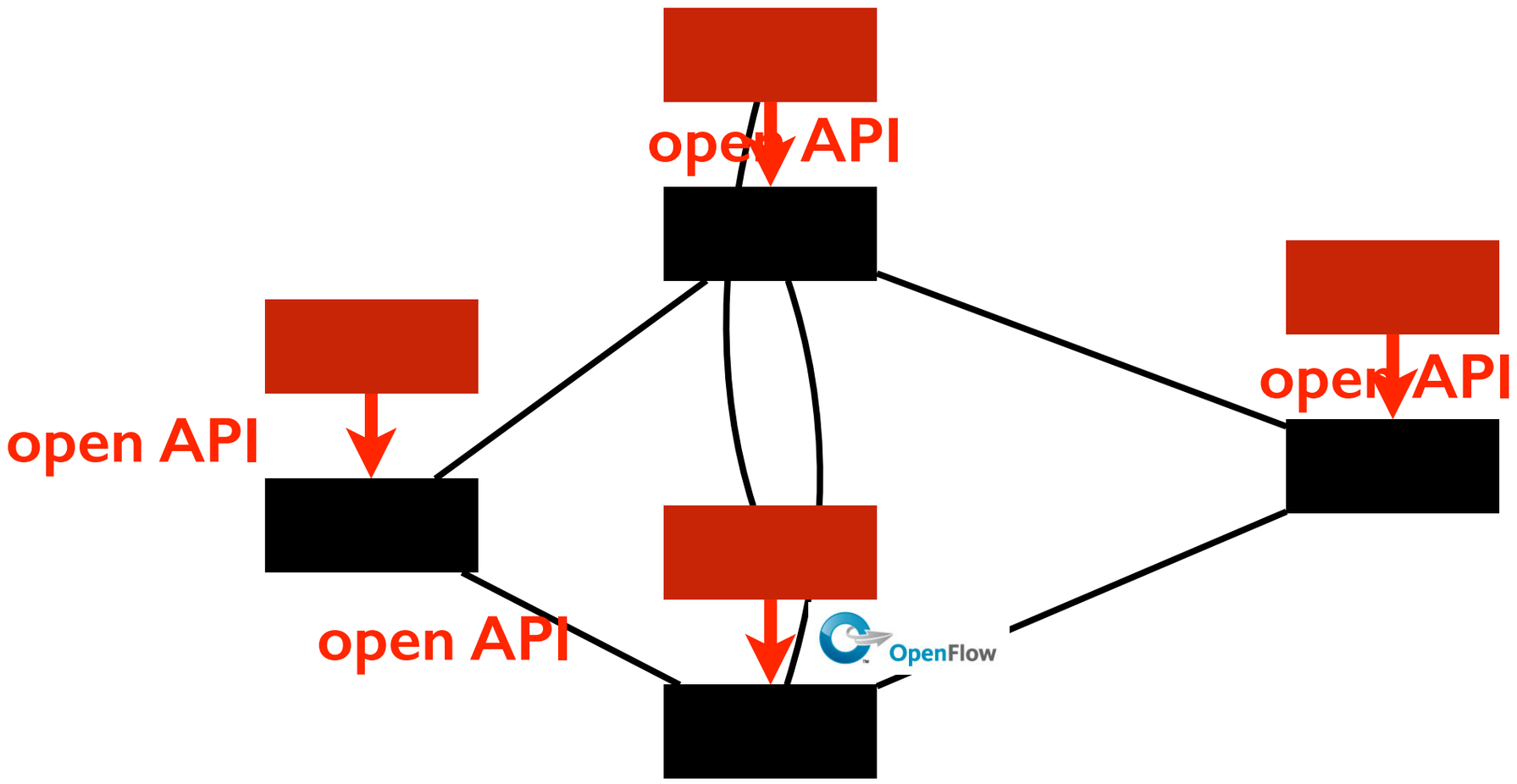


1. src=1.2.*.*, dest=3.4.5.* → drop
2. src = *.*.*.*, dest=3.4.*.* → forward(2)
3. src=10.1.2.3, dest=*.*.*.* → send to controller

Nick McKeown., et al. "OpenFlow: enabling innovation in campus networks"

open dataplane interface

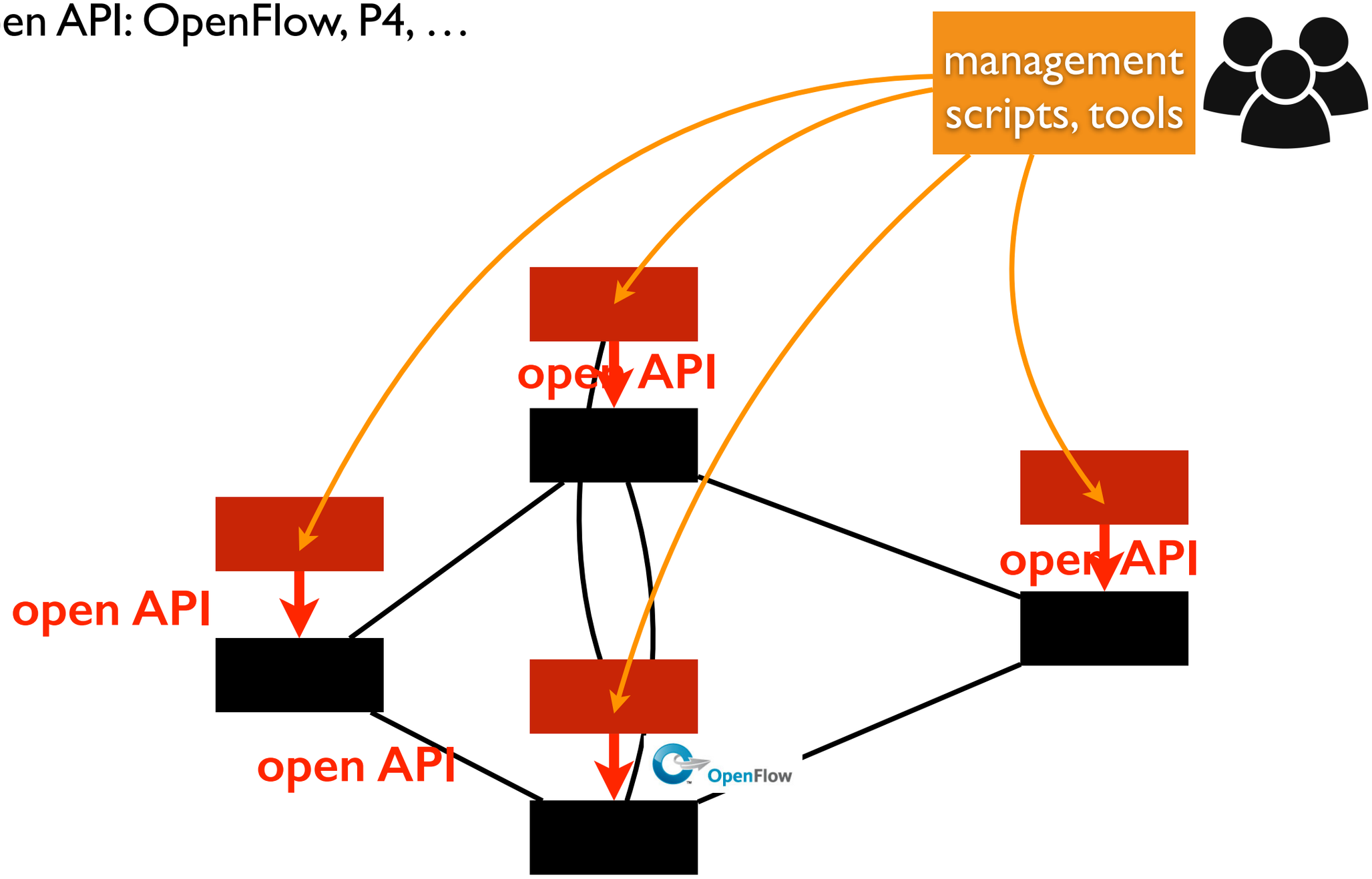
open API: OpenFlow, P4, …



P. Bosshart., et al. P4: Programming Protocol-Independent Packet Processors

open dataplane interface

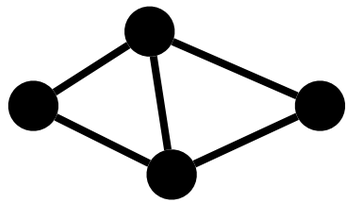
open API: OpenFlow, P4, …

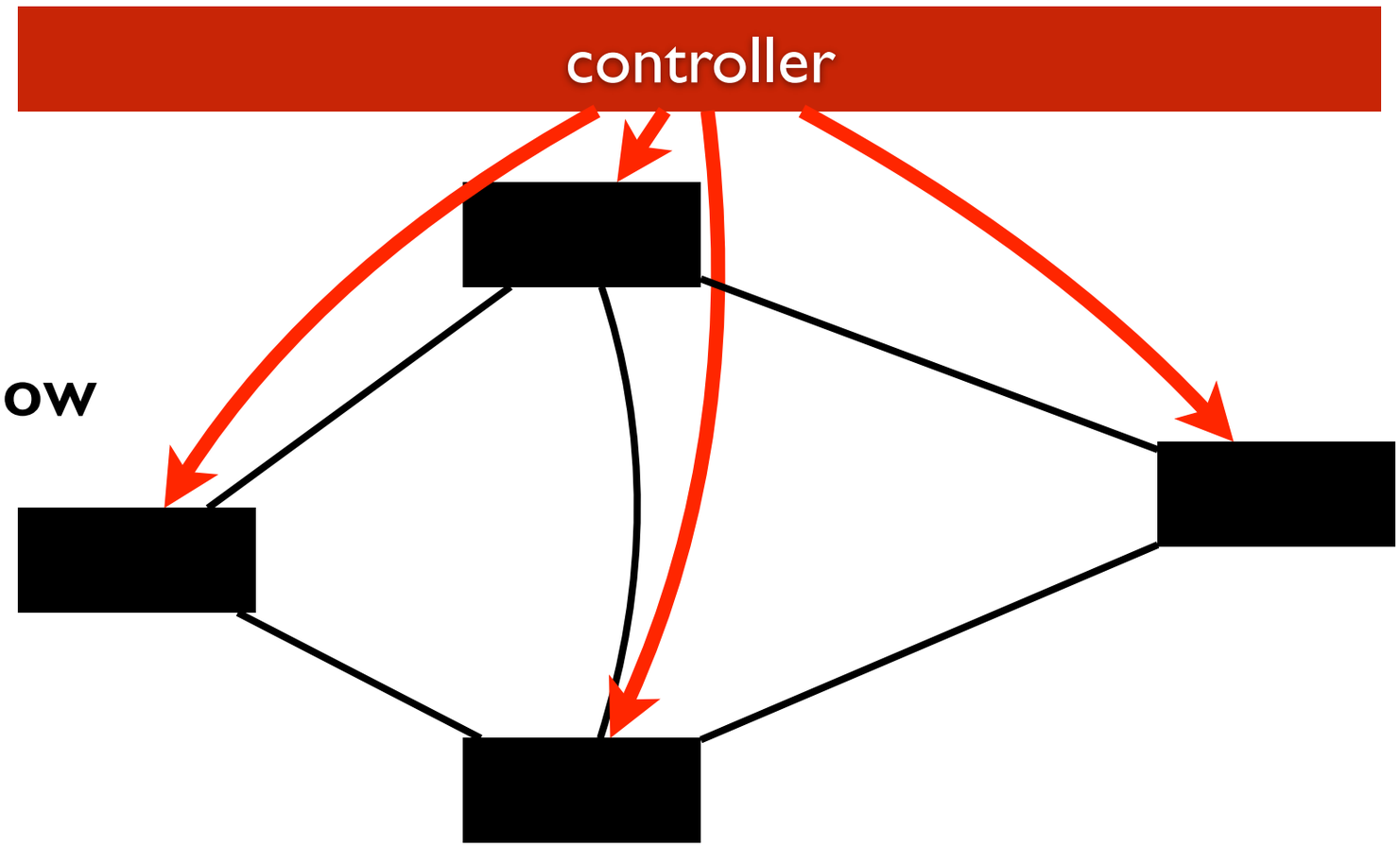


P. Bosshart., et al. P4: Programming Protocol-Independent Packet Processors

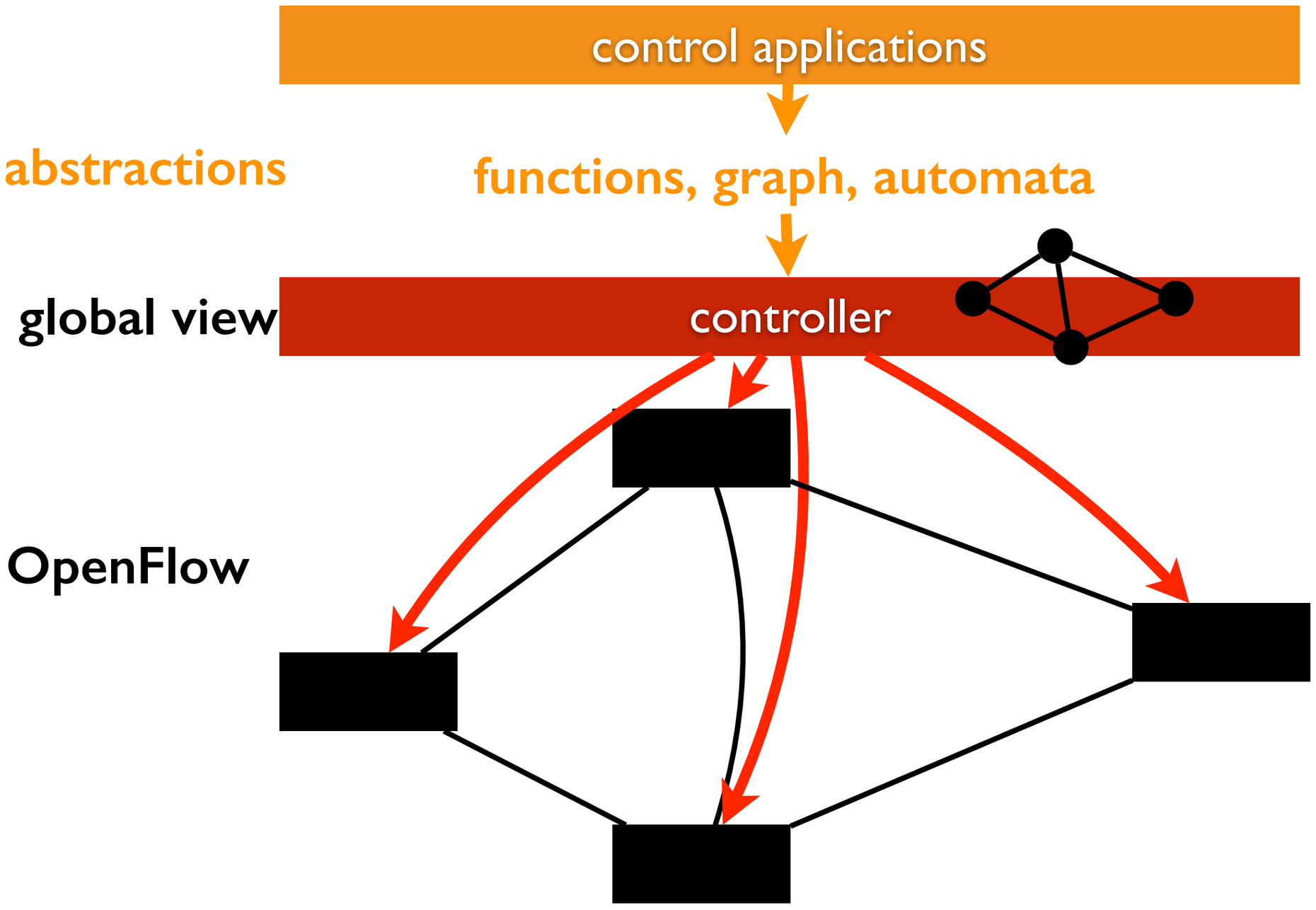
(logically) centralized controller

global network view

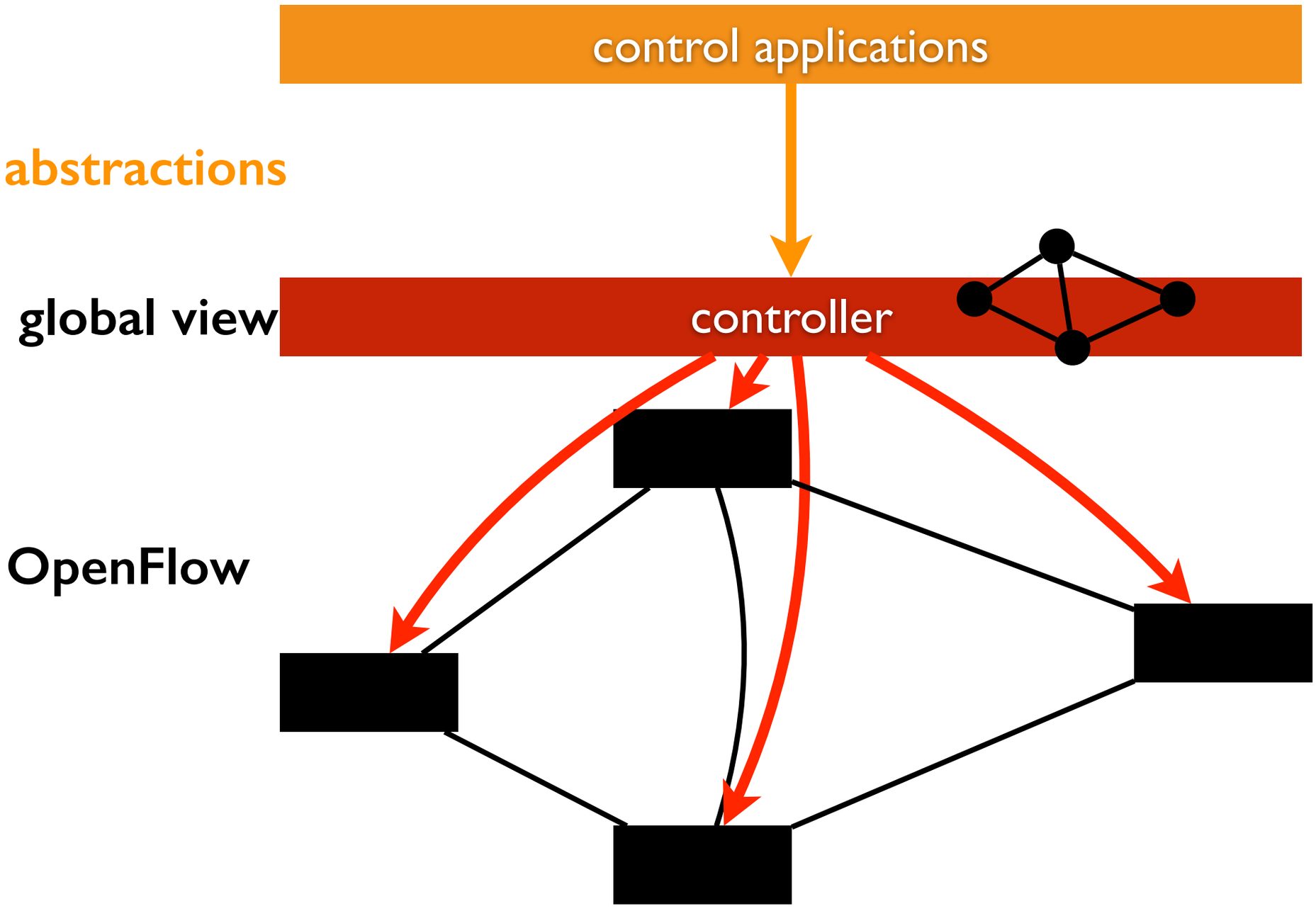




higher-level abstractions



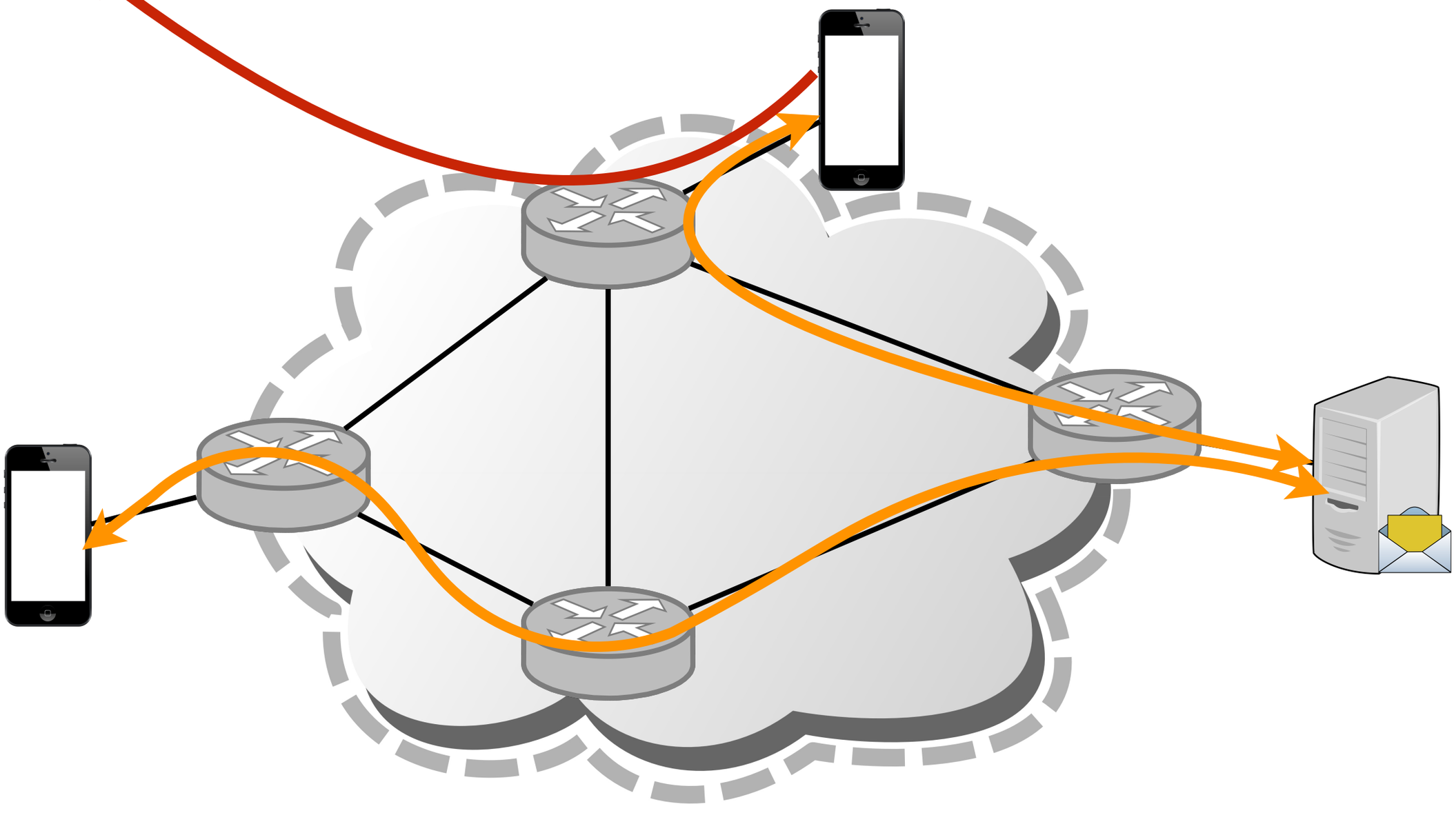
protocols ➔ applications



application: seamless mobility

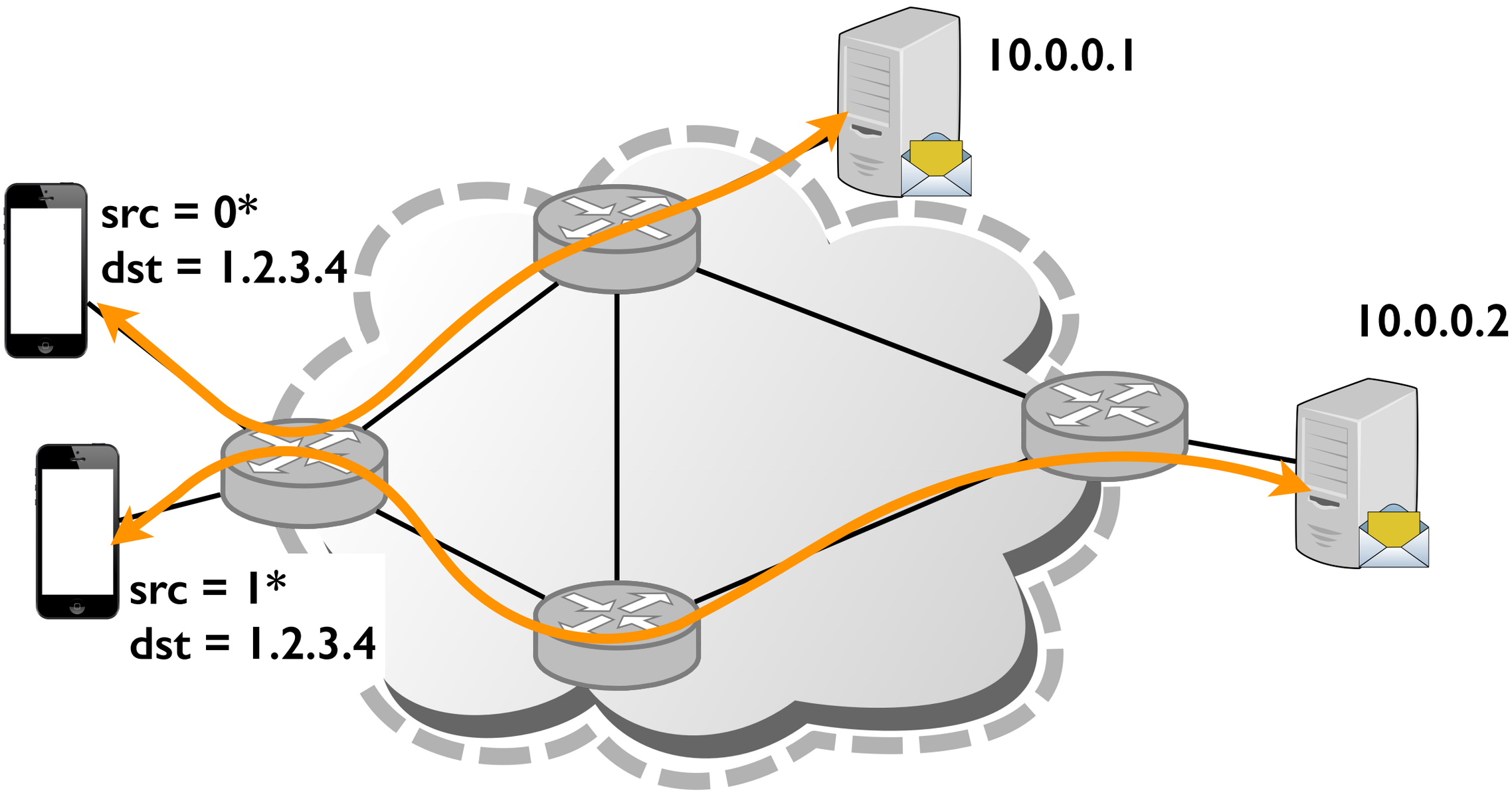
app

- See host sending traffic at new location
- Modify rules to reroute the traffic



application: server load balancing

- pre-install load-balancing policy
- split traffic based on source IP

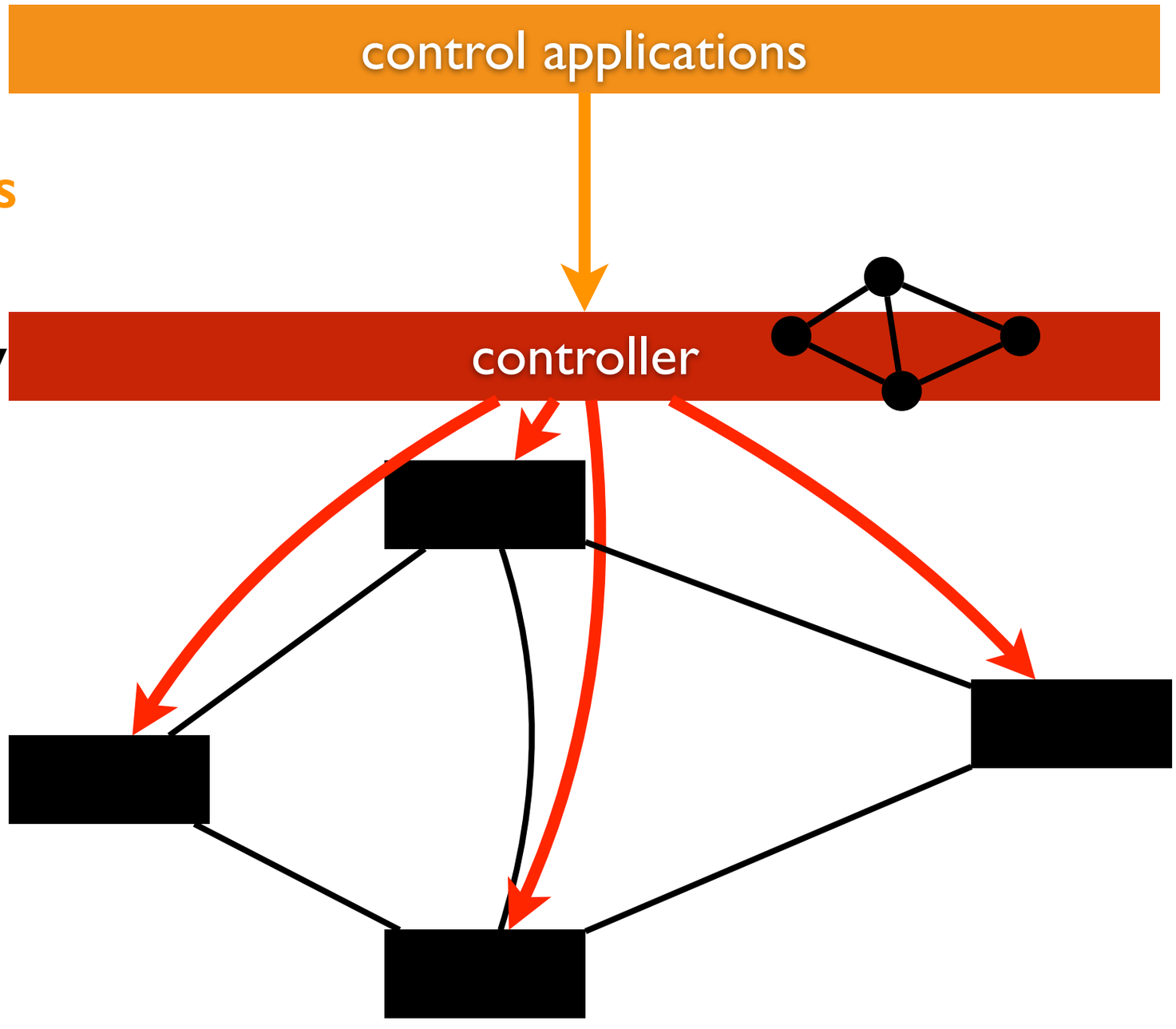


protocols ➔ applications

abstractions

global view

OpenFlow

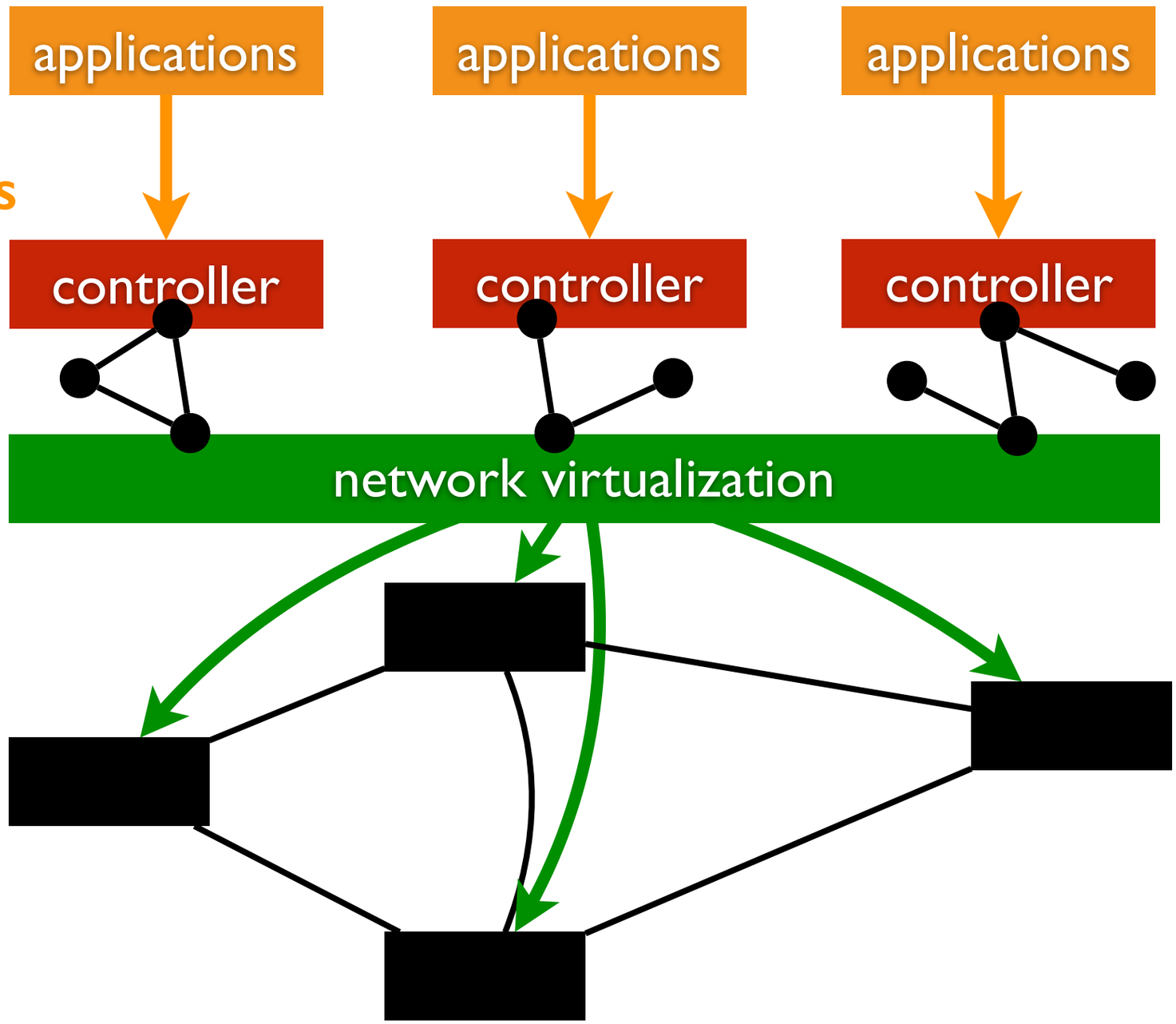


network virtualization

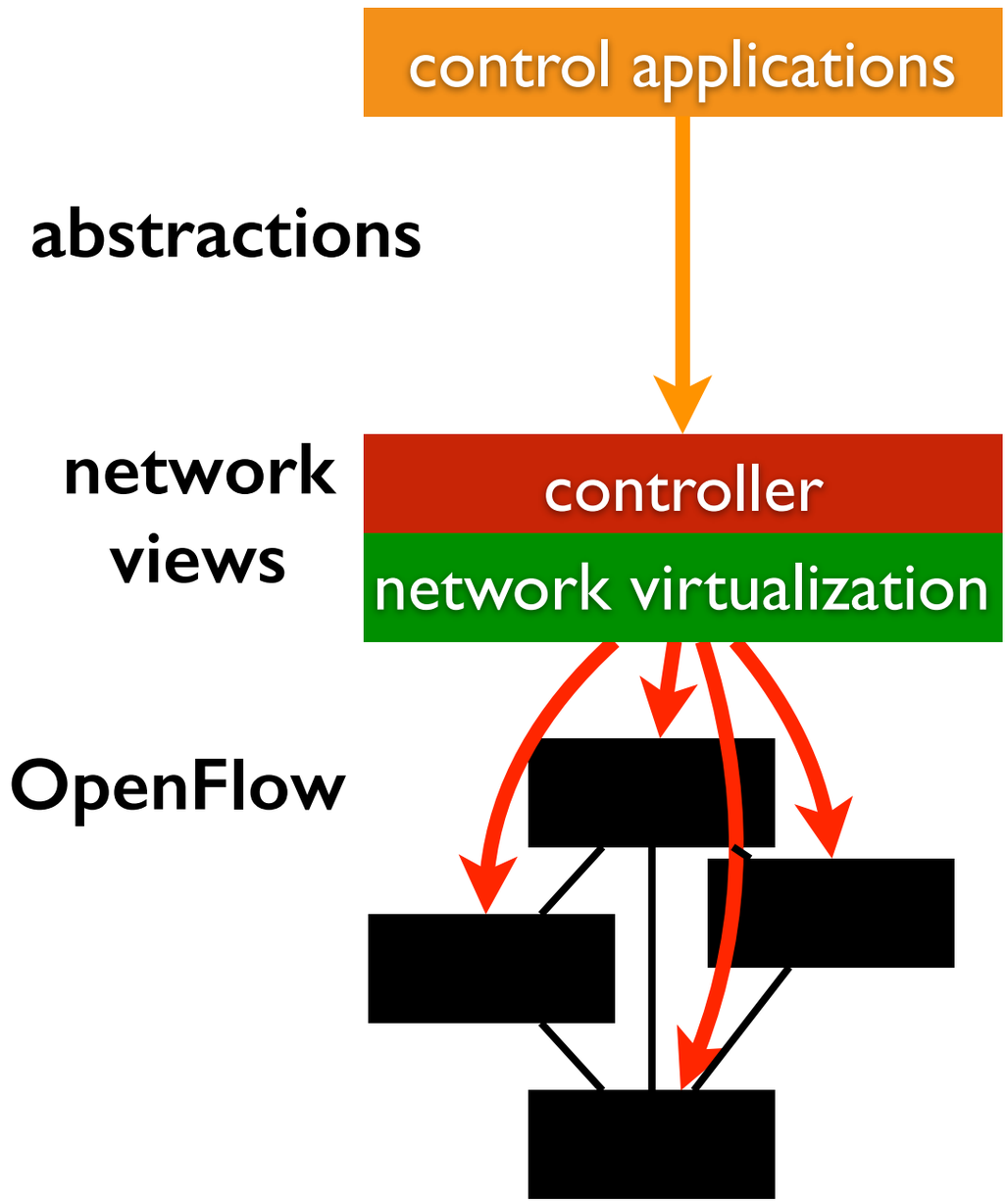
abstractions

virtual Net

OpenFlow



recap: SDN technologies



supporting technologies

- central network control
- programmability
- network virtualization

benefits

- simplified operation with direct, network-wide control
- cost reduction with open hardware

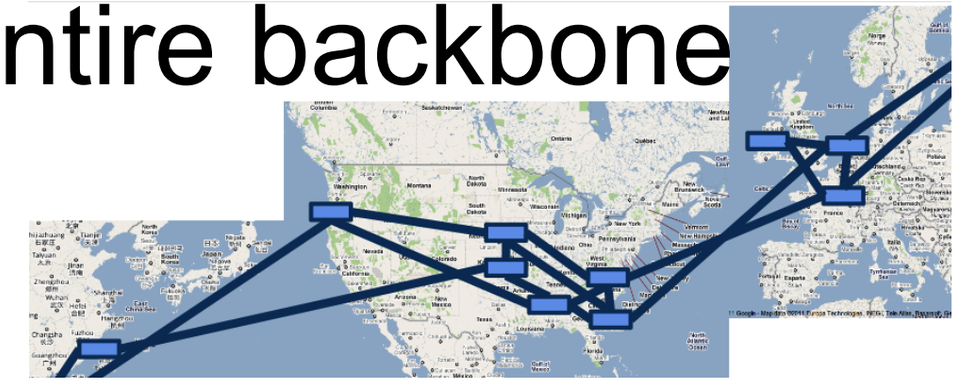
a major trend in networking

OPEN NETWORKING FOUNDATION

Deutsche Telekom · facebook · Goldman Sachs · Google · Microsoft · NTT Communications · verizon · YAHOO!

Google

Entire backbone



runs on SDN

nicira

Bought for **$1.2 x 10^9**
(mostly cash)

an opportunity to rethink

disciplines

- how should future networks be?
  - designed, programmed, operated, managed …
- what are the right abstractions?
  - single task, integration, updates …

structure of the course

syllabus

introduction

- review, historical evolution

SDN basics

- centralized control, programmability, network virtualization

advanced topic

- network function virtualization (NFV)

innovation and application

- verification, synthesis

paper reading

read 2~4 papers for each class (2 lectures)

- recent research papers on SDN
- basis for discussion in class

write review for 2 (1 page each)

- summary (problem, solution)
- what you like, to improve, to avoid

one in class oral presentation

- conference style: 20min talk + 5min Q&A

to do

- read "How to Read" on today's syllabus

lightweight assignments

programming assignments

- Mininet platform
- POX, Ravel controller

assignments are *not* graded

collaboration policy

- can freely collaborate with others

will help your course project

course project

final research project

- work alone or in pair
- your own topic, or from a list we suggest

schedule

- talk to me (and others) about project ideas
- 5pm Thur Oct 20: short proposal due
- 5pm Sun December 18: written report due
- last week: short oral presentation

grading

- 0% programming assignments
- 30% class participation (discussion, presentation)
- 40% paper reviews
- 30% course project (paper, talk)

to do

next steps

- join the Piazza site
  - piazza.com/temple/fall2016/5590
- course web site
  - cis-linux1.temple.edu/~tug29203/16-5590/