(Ir)relevance reasoning for software-defined network

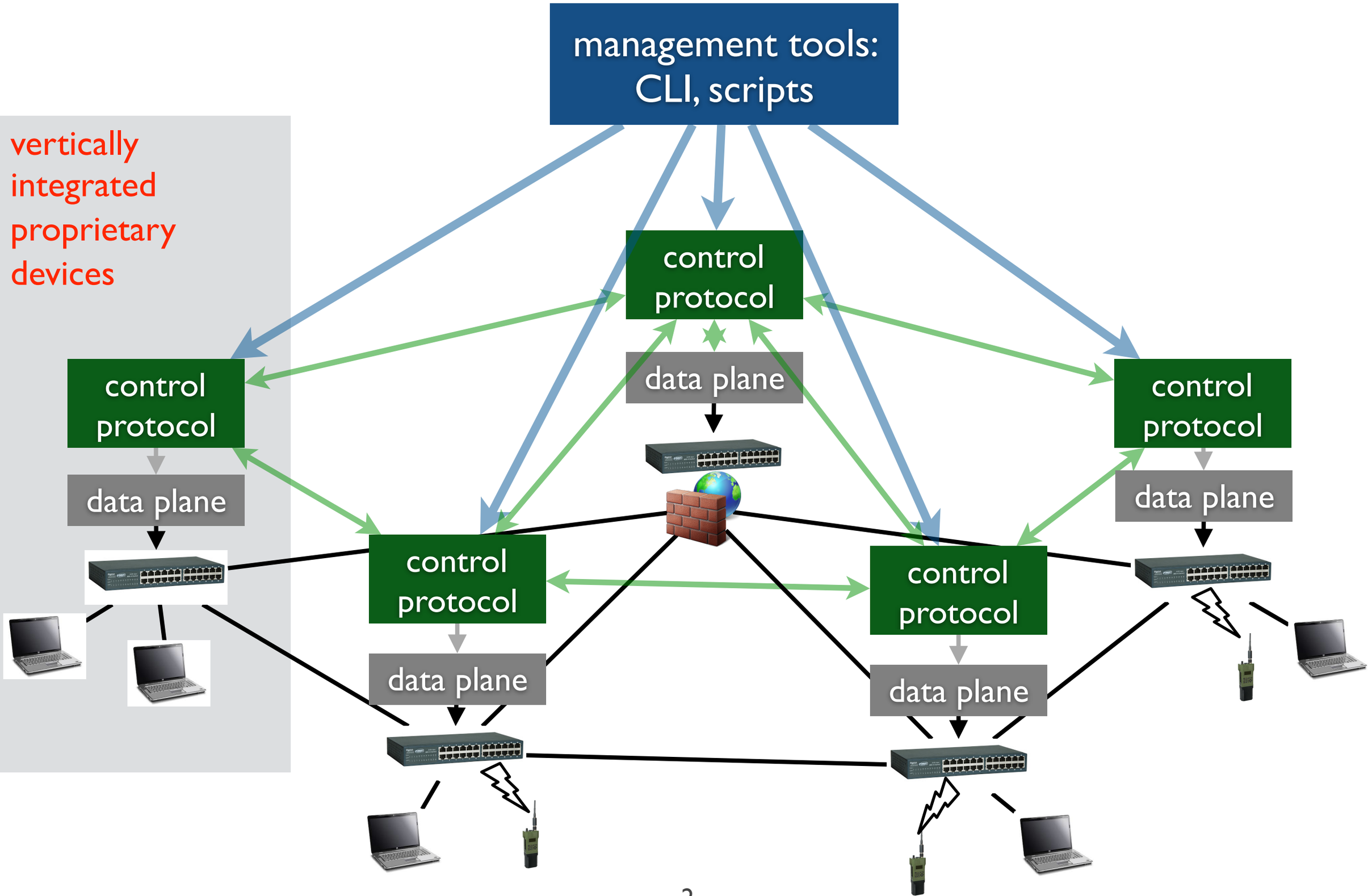
Anduo Wang[*] Jason Croft[†] Xueyuan Mei[†]
Matthew Caesar[†] Brighten Godfrey[†]

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

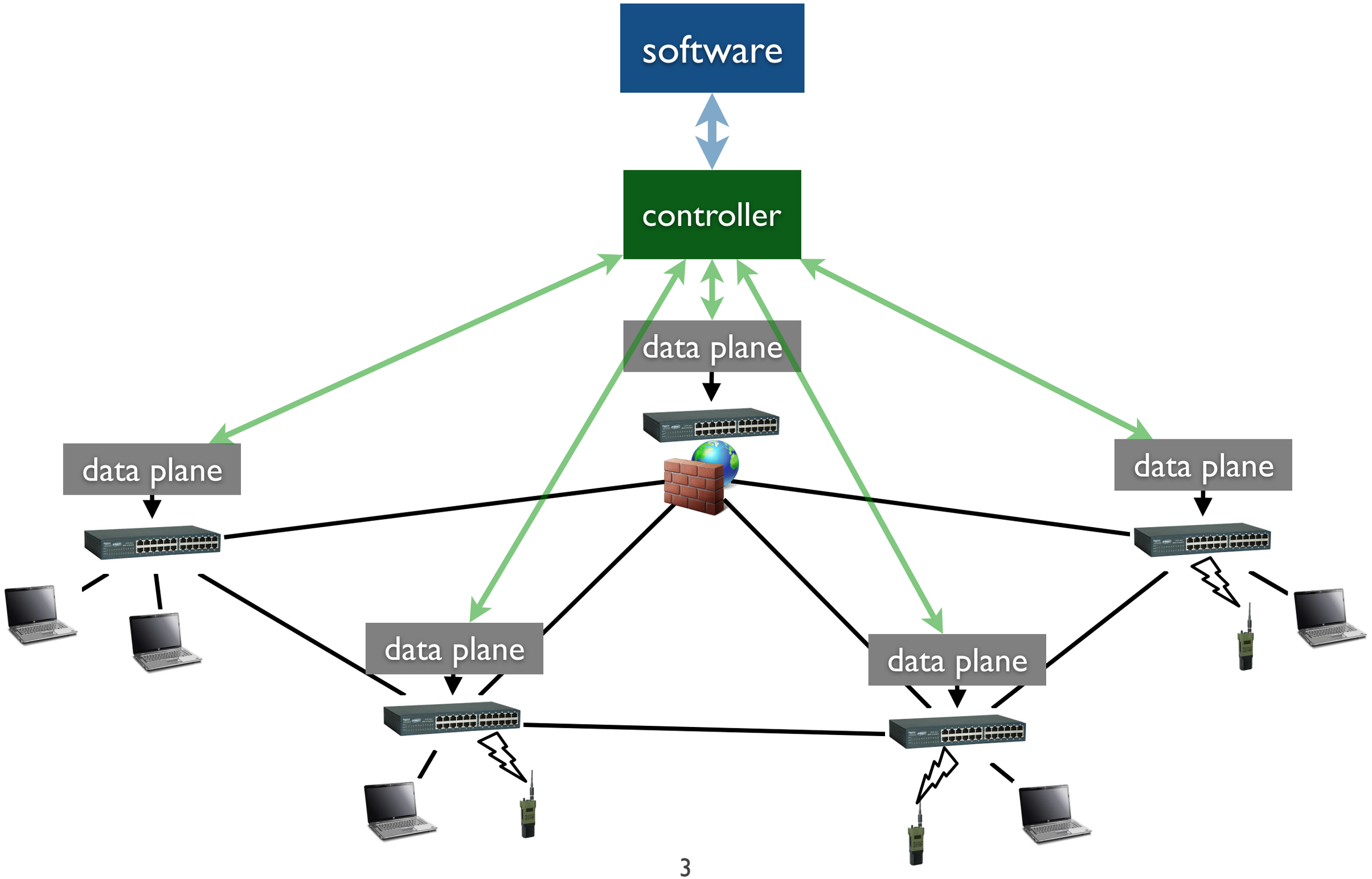
ExCAPE PI Meeting, May 9, 2016

traditional networking

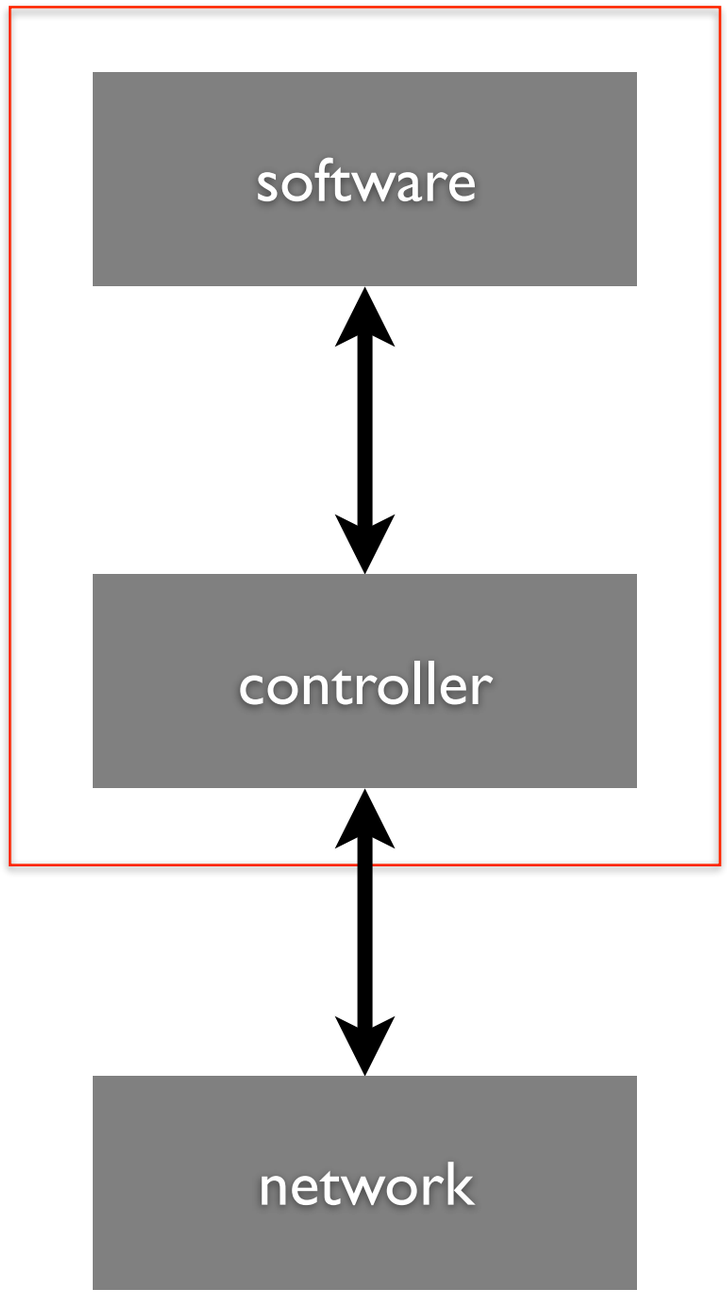
vertically integrated proprietary devices



software-defined networking (SDN)

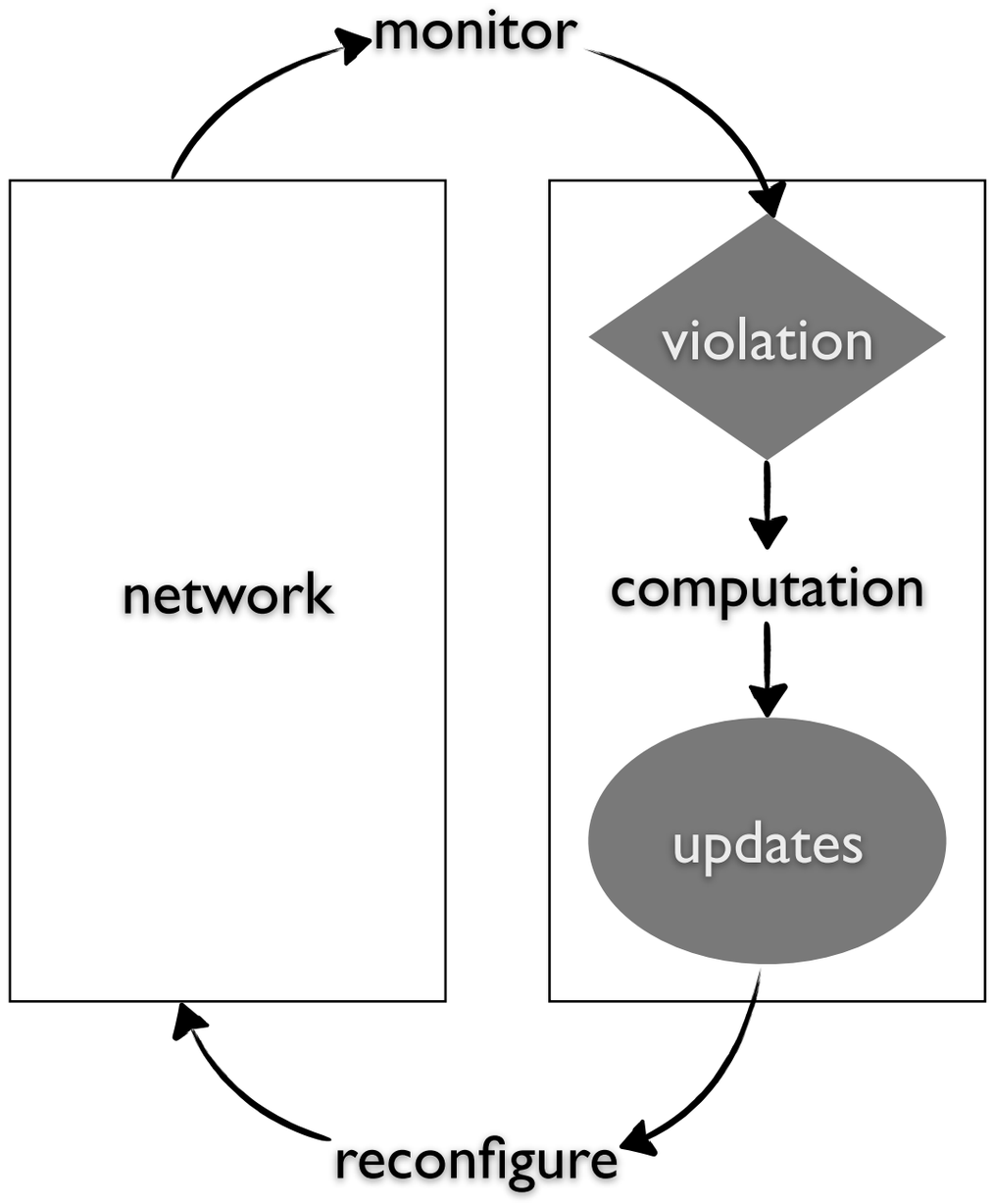


software-defined networking (SDN)

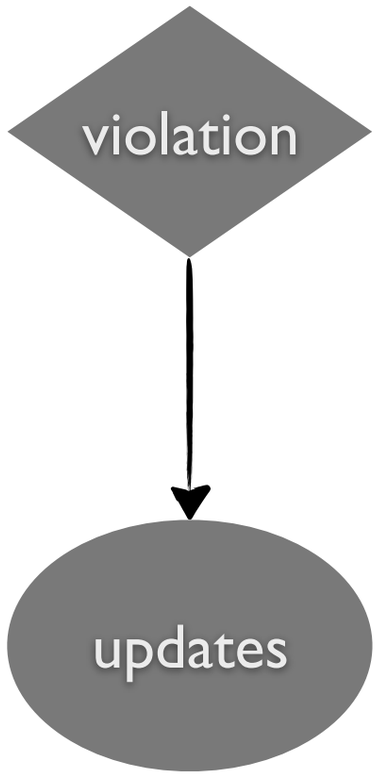


SDN moves complexity to
control softwares:
an opportunity and challenge

SDN control software

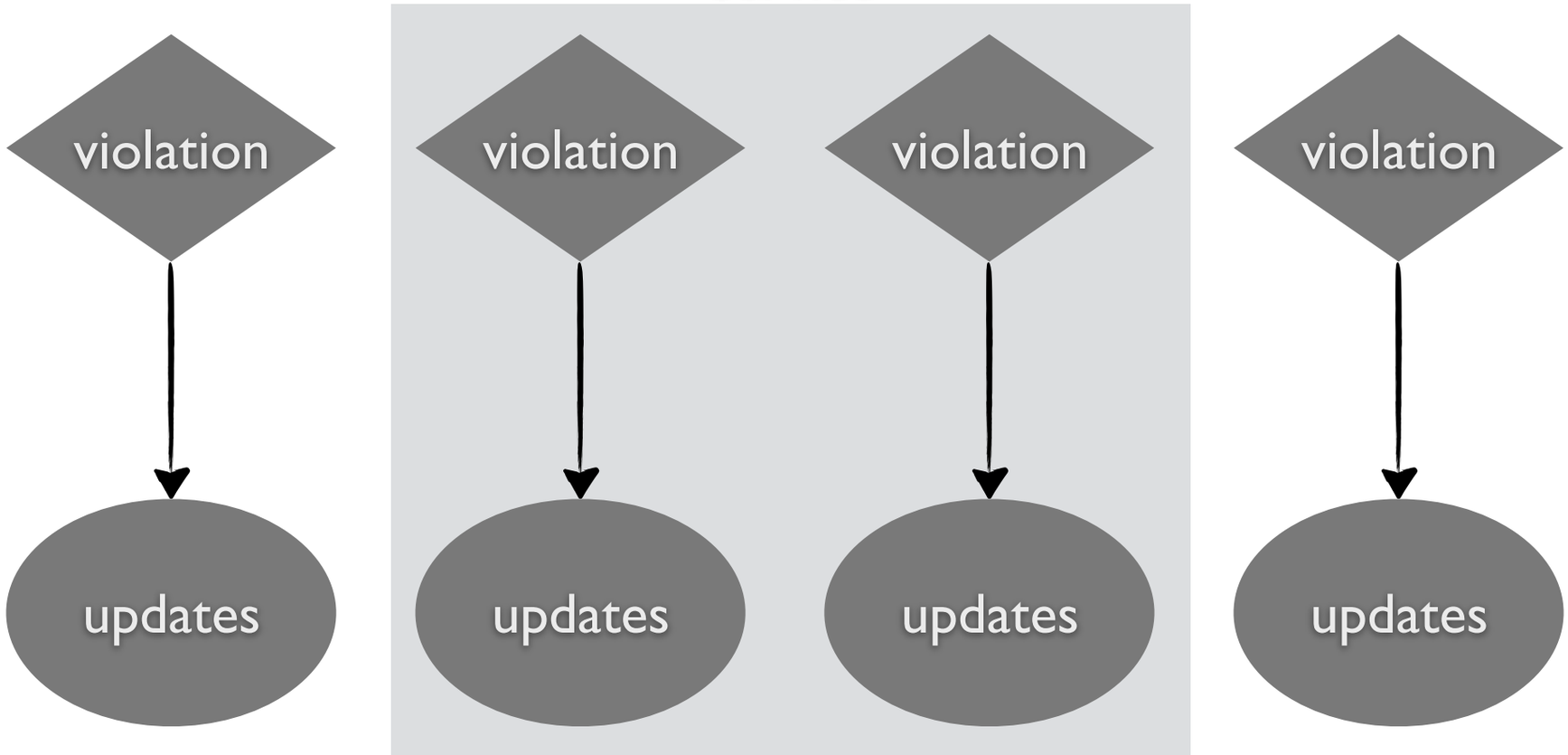


SDN control software

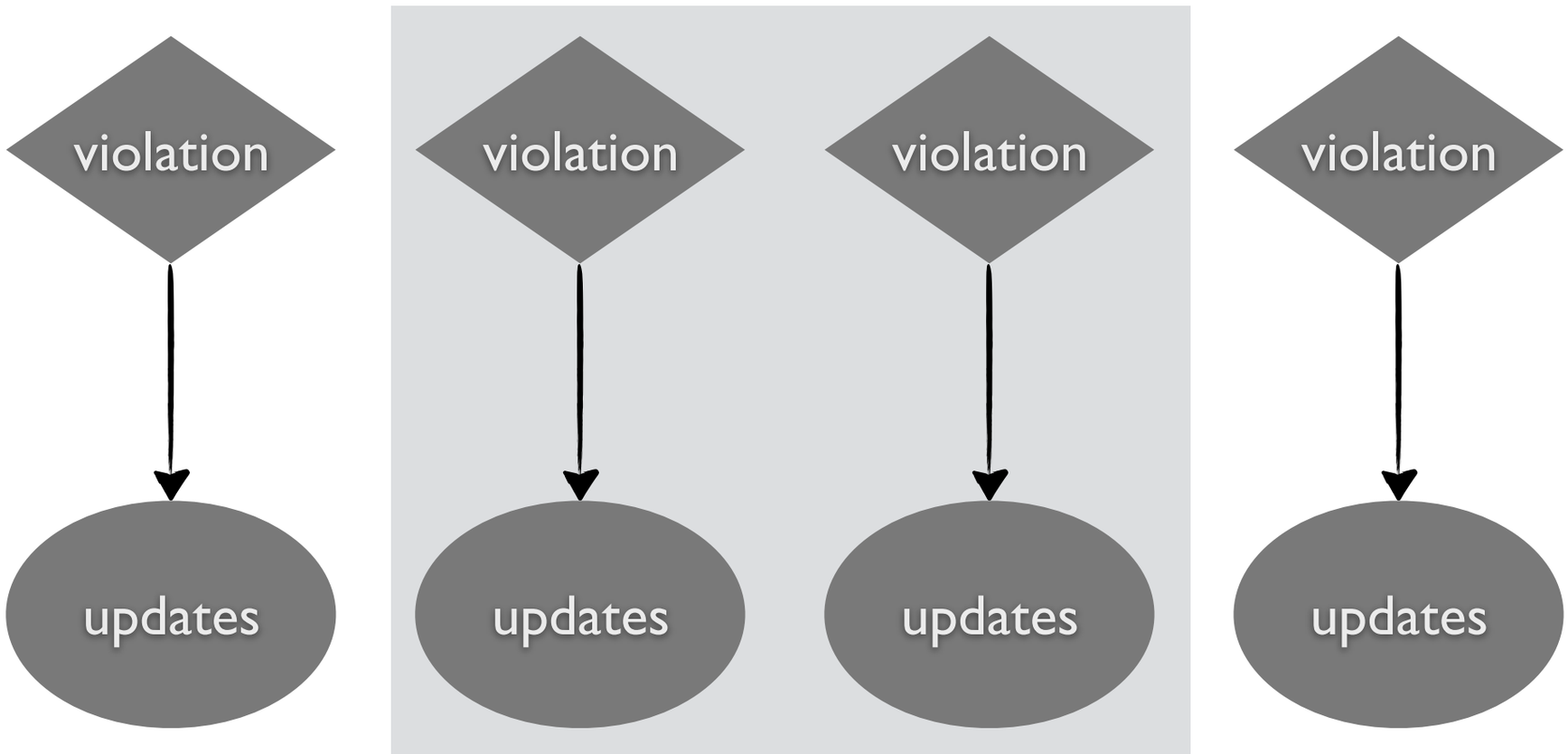


SDN control software

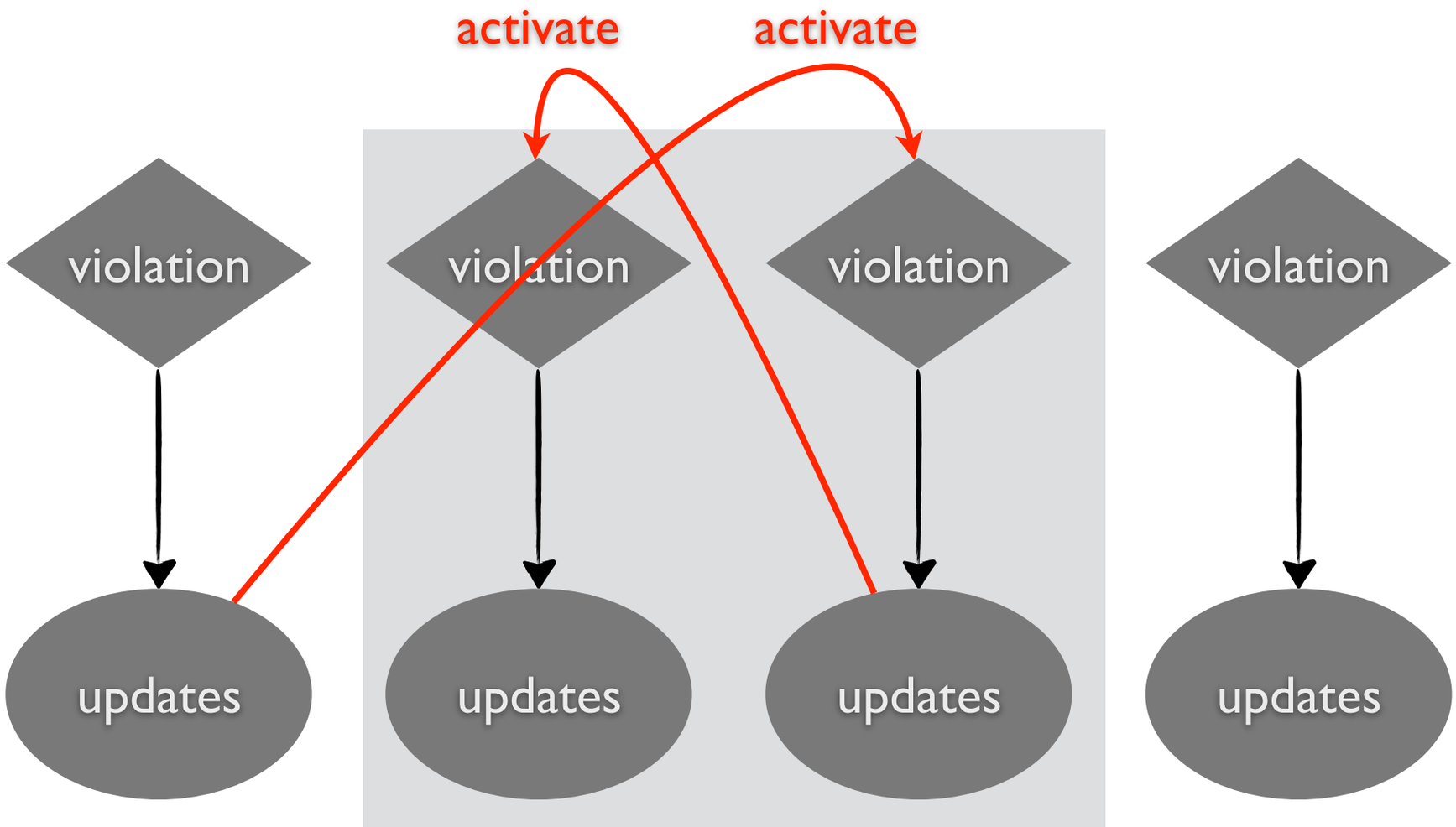
grouped into application /
component /
module (this talk) /
service



SDN control software



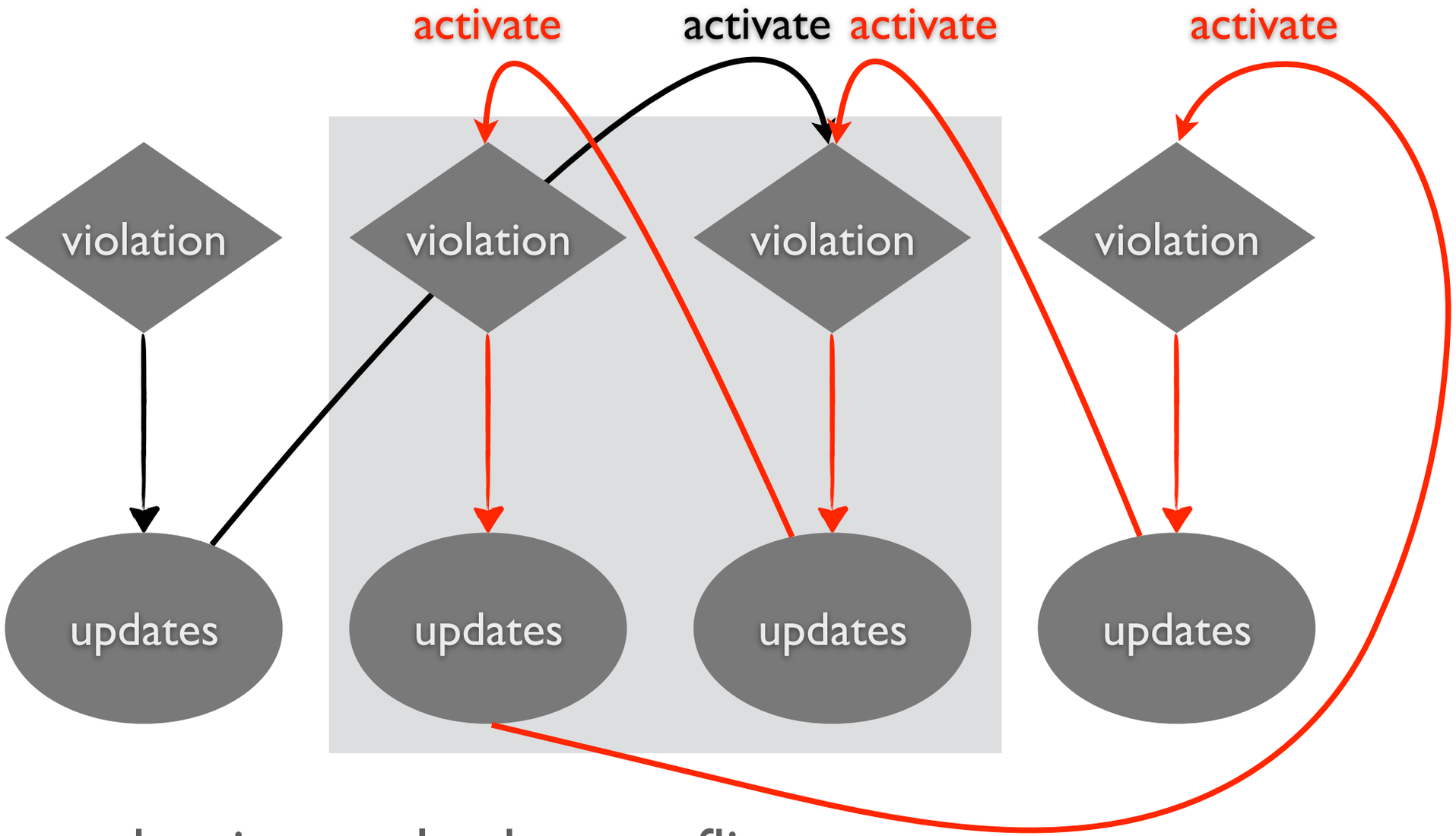
SDN control software



dependency within and across modules

- **modular programming**: instrumentation by master programs
- **limitation**: manual, requires understanding of module internals

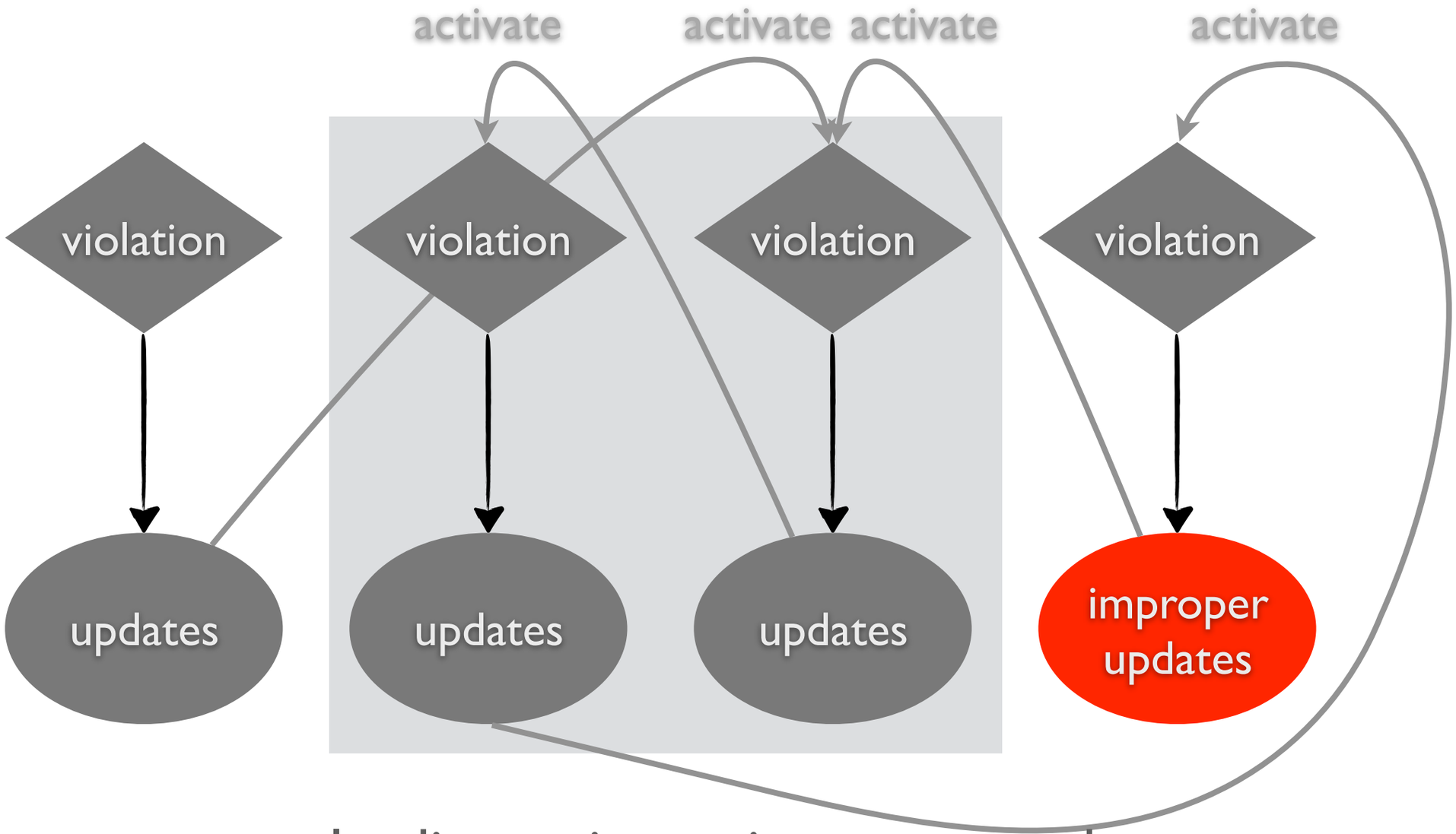
SDN control software



multiple dependencies can lead to conflict

- **conflict resolution:** application-/module- level priority
- **limitation**: coarse-grained, manual

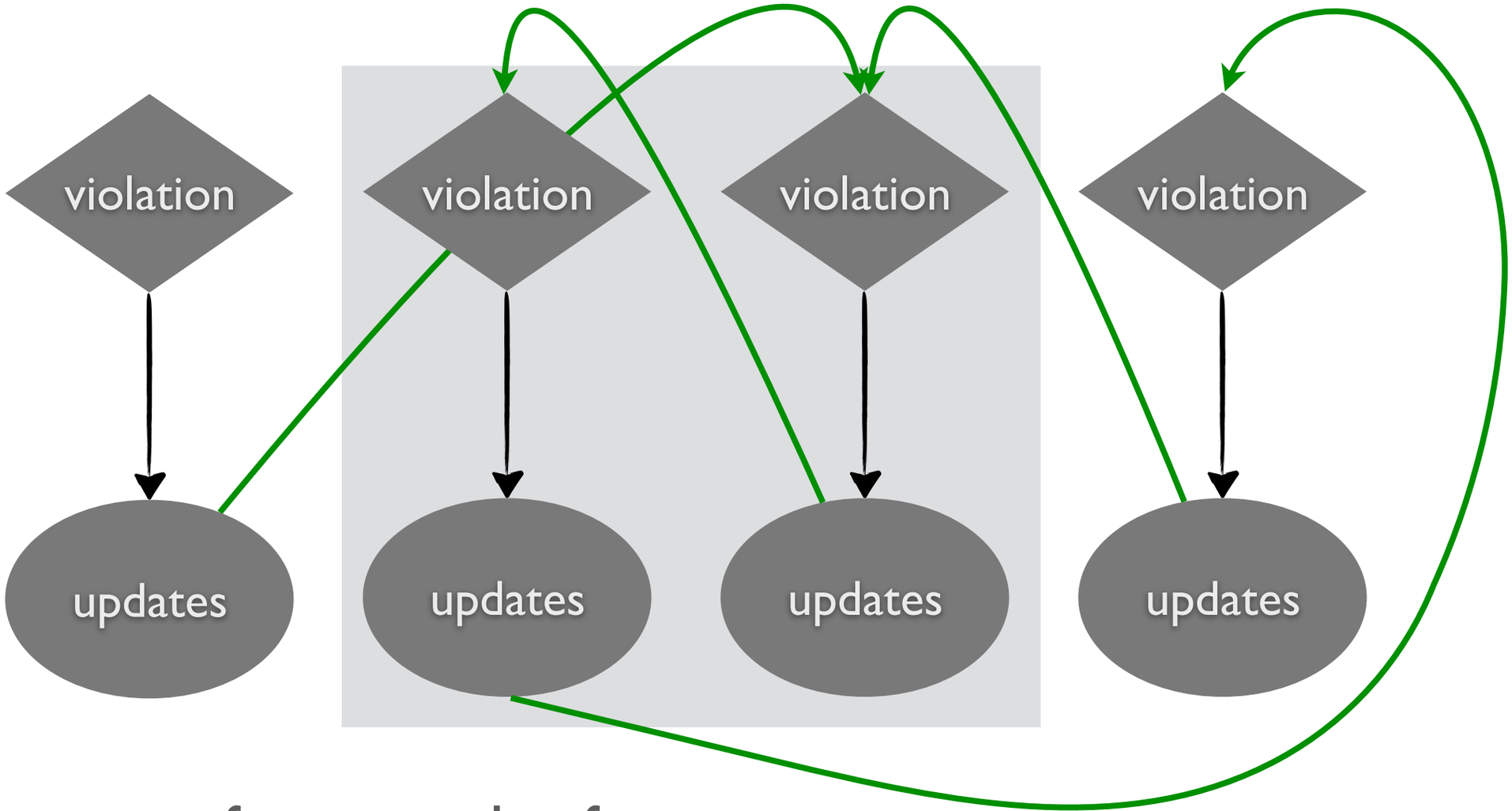
SDN control software



updates may go wrong, leading to inconsistent network states

- **debugging and verification**: detect inconsistent states, locate events leading to an error
- **limitation**: post-mortem, not revealing root causes in control software

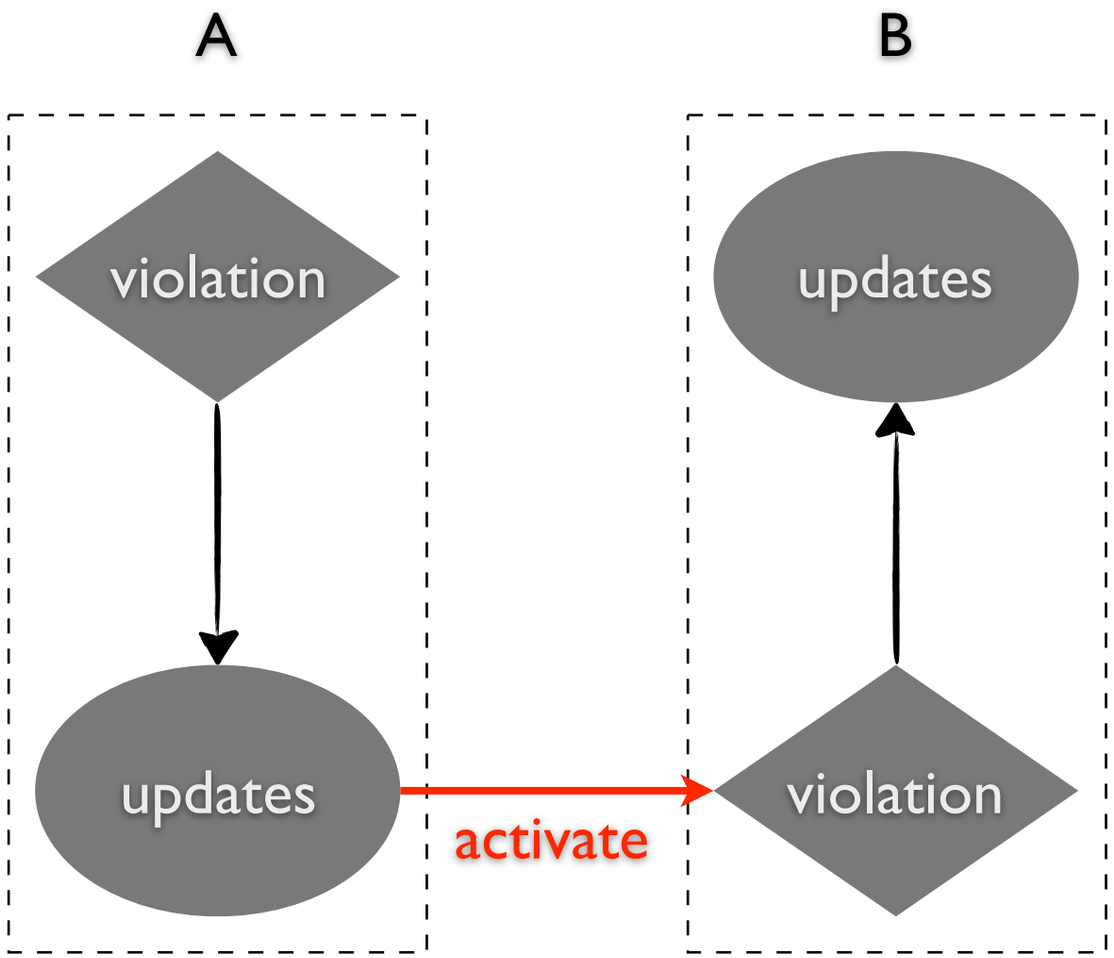
our approach: automated reasoning



reasoning support for control software

- **automated:** leveraging formal tool (SMT solver)
- **finer-grained:** update-level
- **static:** prior-to deployment, reveal interactions between / within control modules

dependency

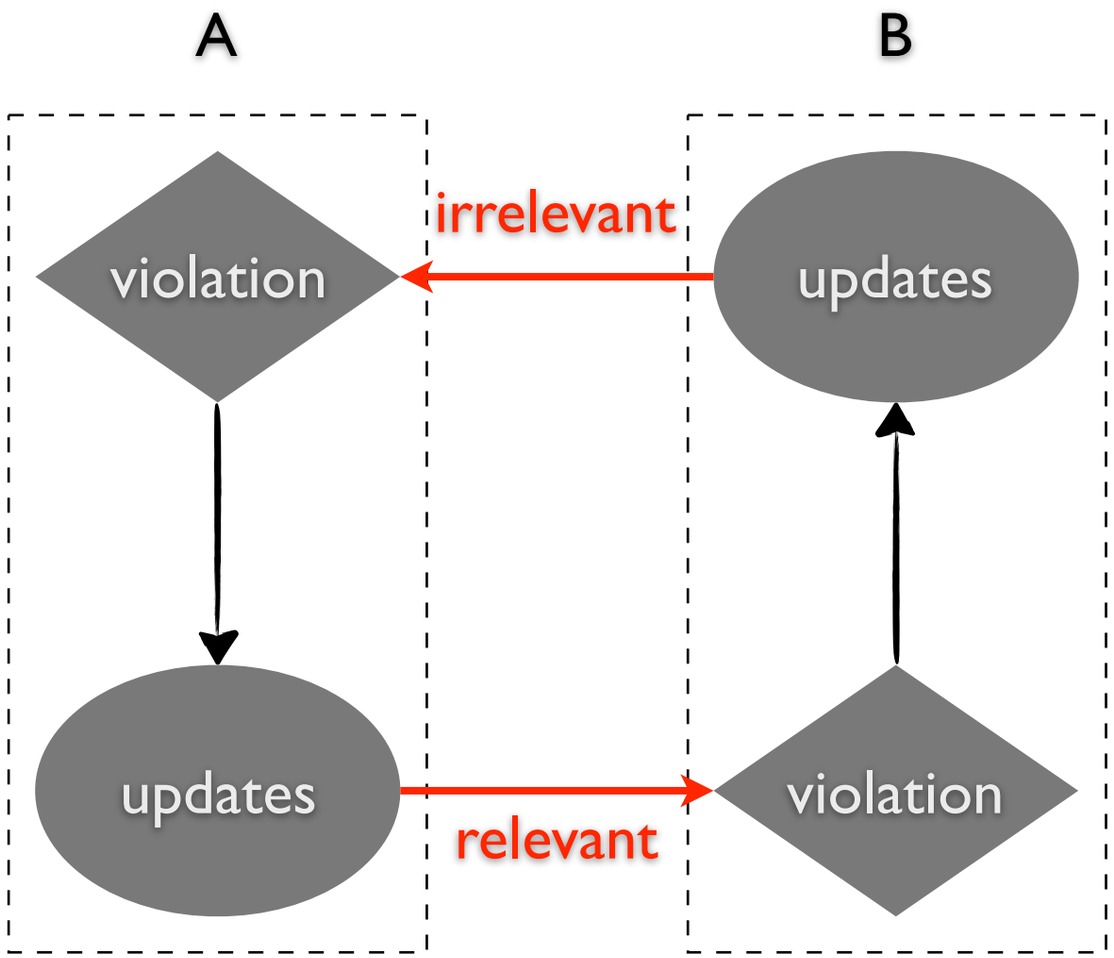


we say A depends on B, if

(1) A update can activate B

(2) B update will never activate A

dependency by (ir)relevance reasoning

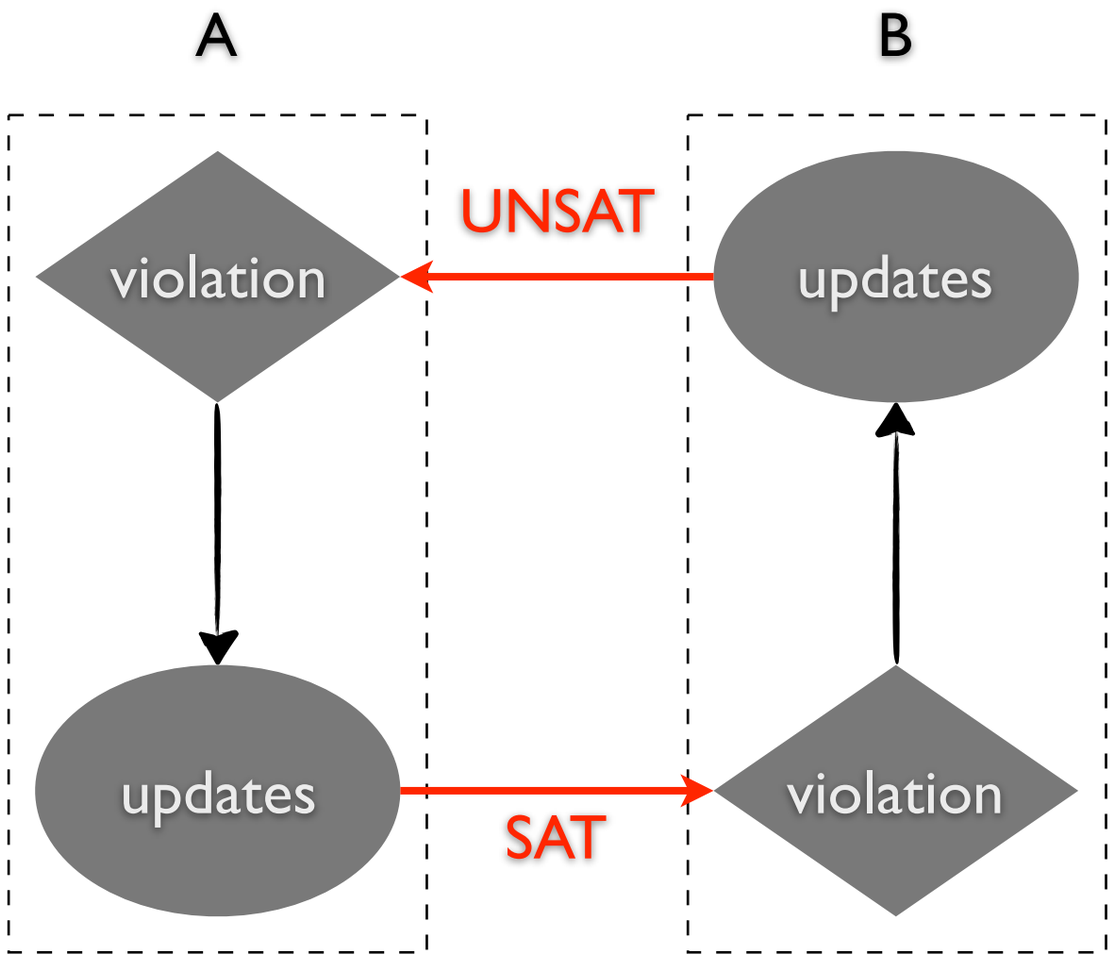


A depends on B, if

(1) A is relevant to B

(2) B is irrelevant to A

(ir)relevance to satisfiability

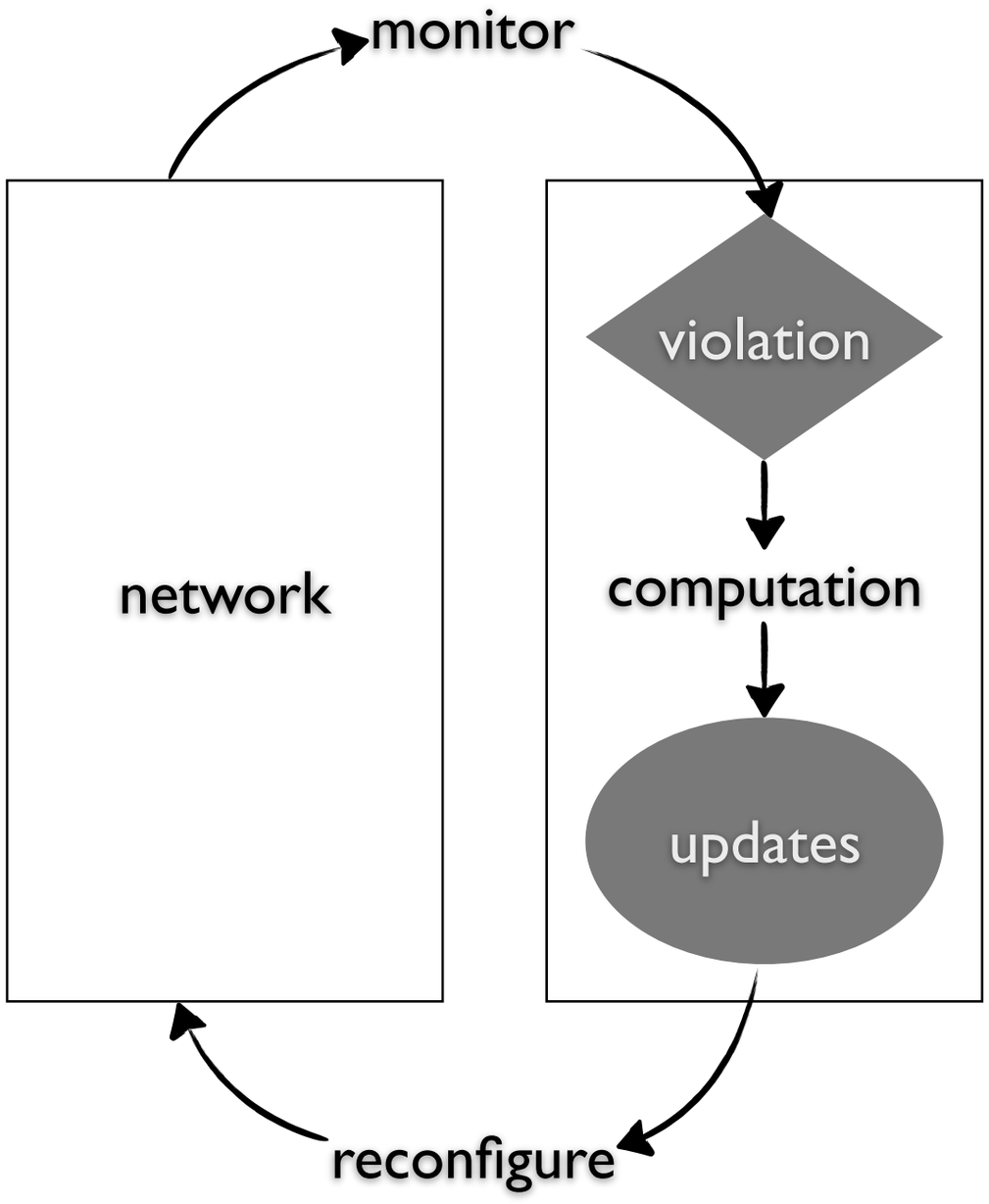


A depends on B, if

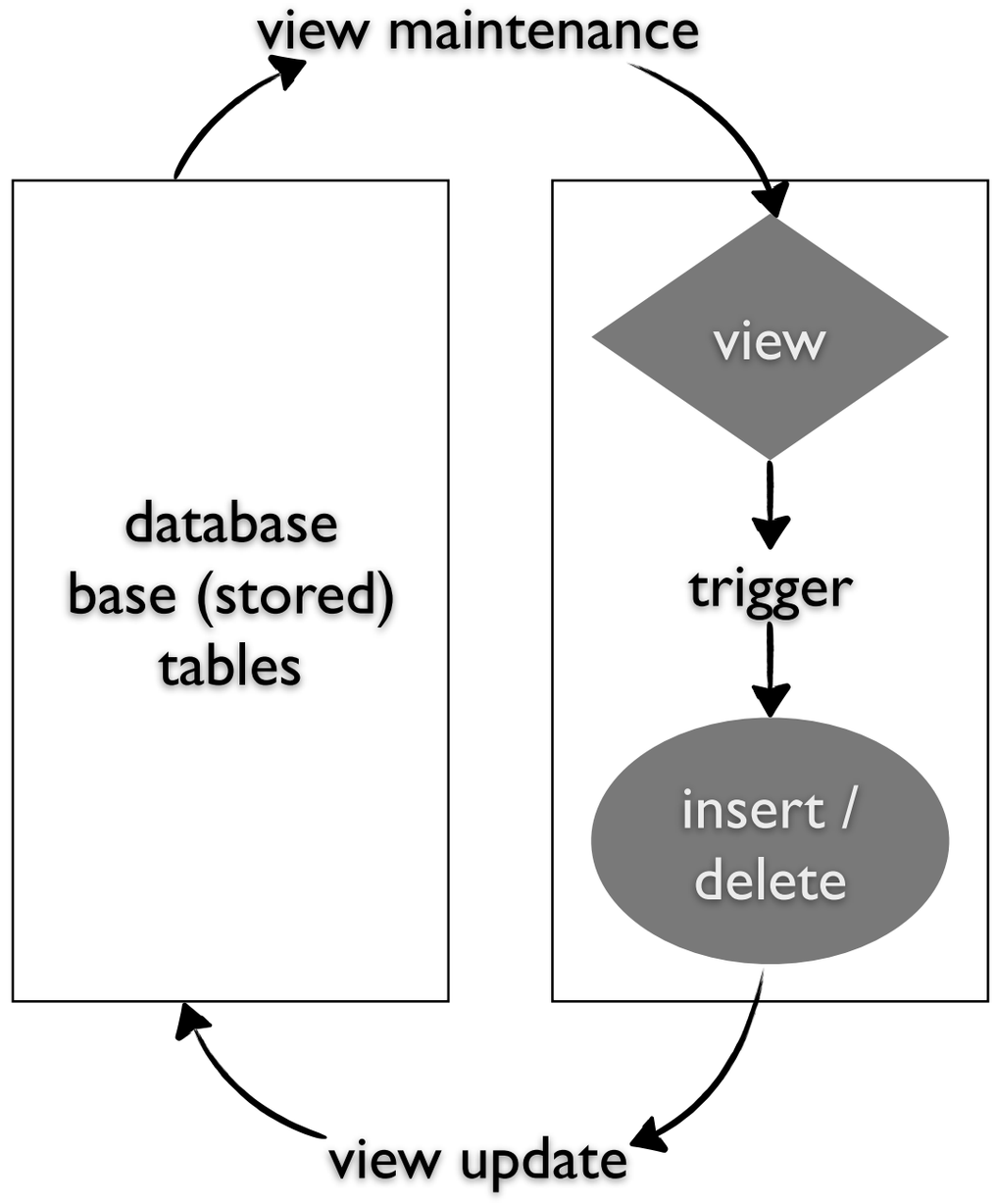
- can find a network state where A update violates B
- no network state exists where B update can ever violate A

formal model

SDN control loop



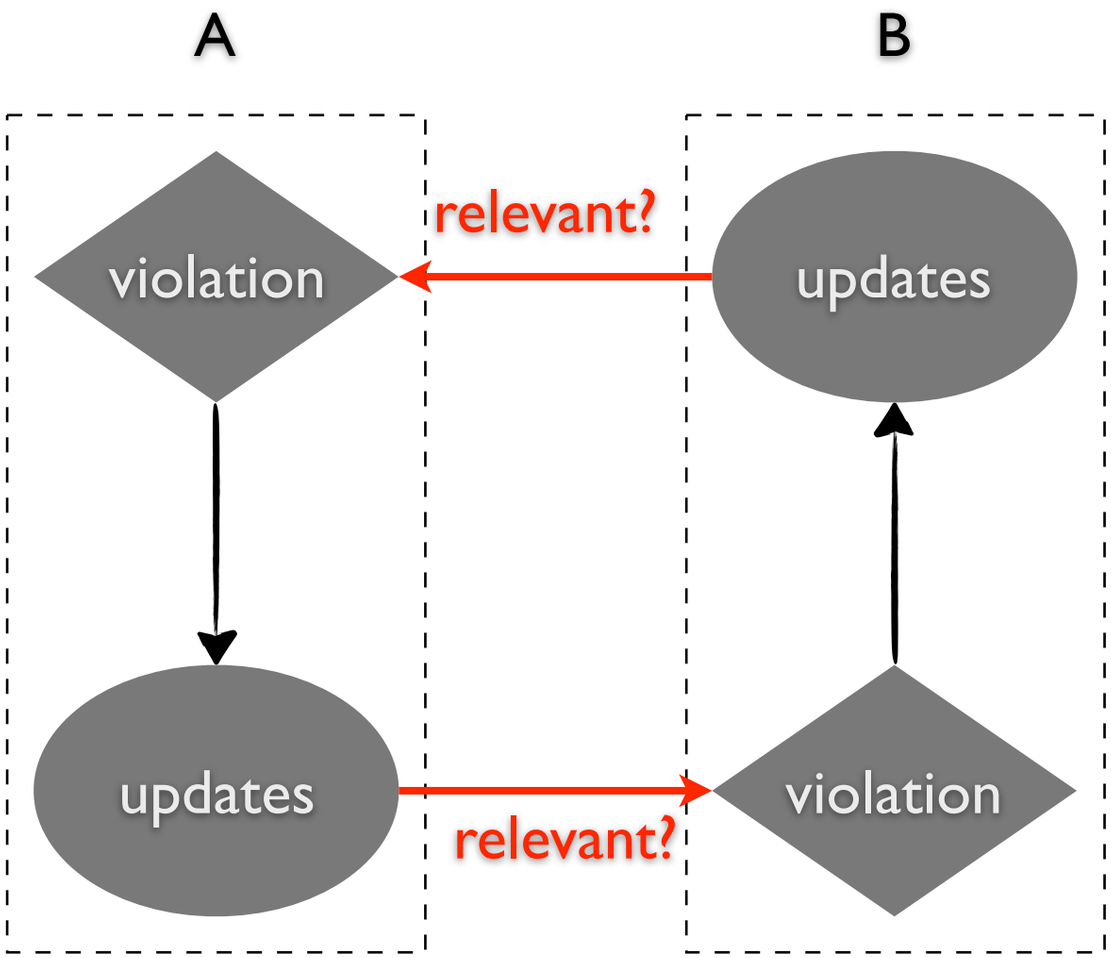
a unified database representation



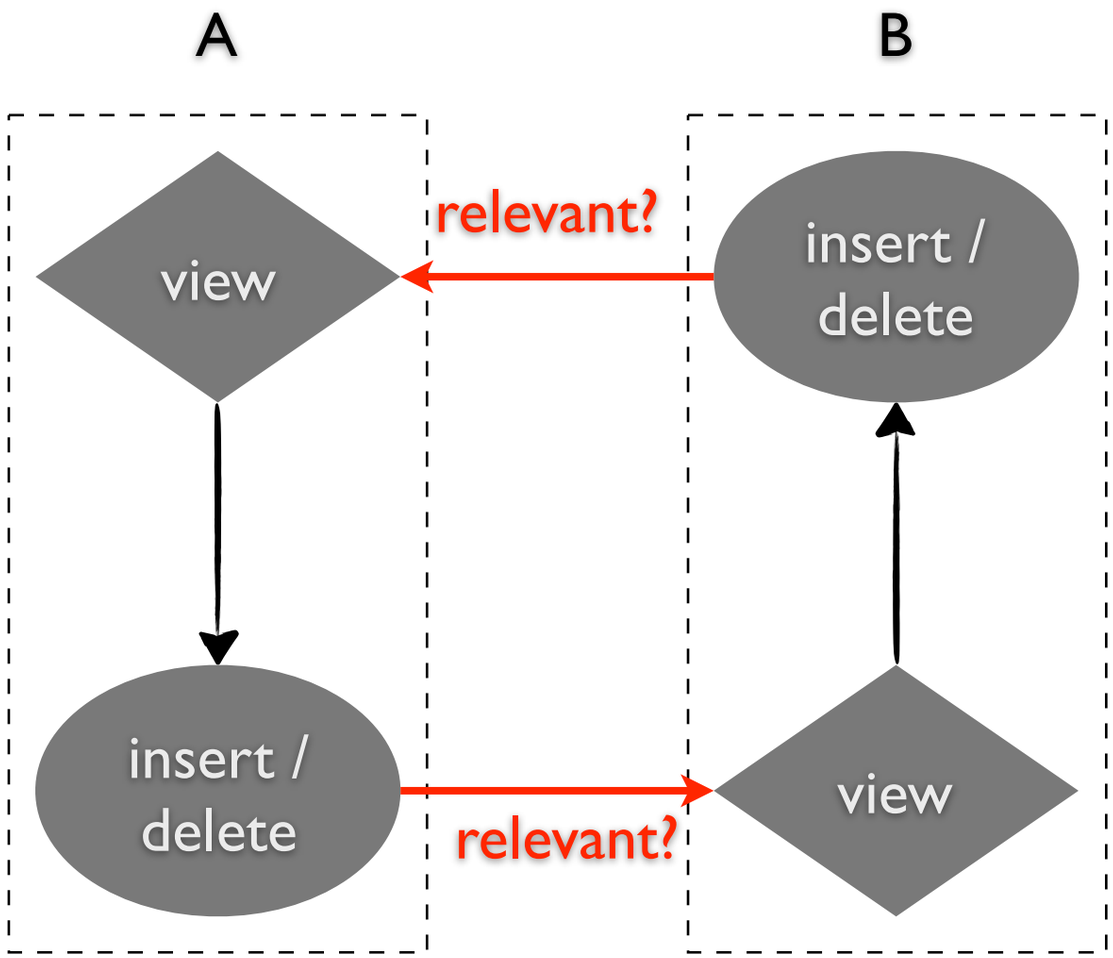
ravel: a database-defined network [SOSR'16]
ravel-net.org

database irrelevance reasoning

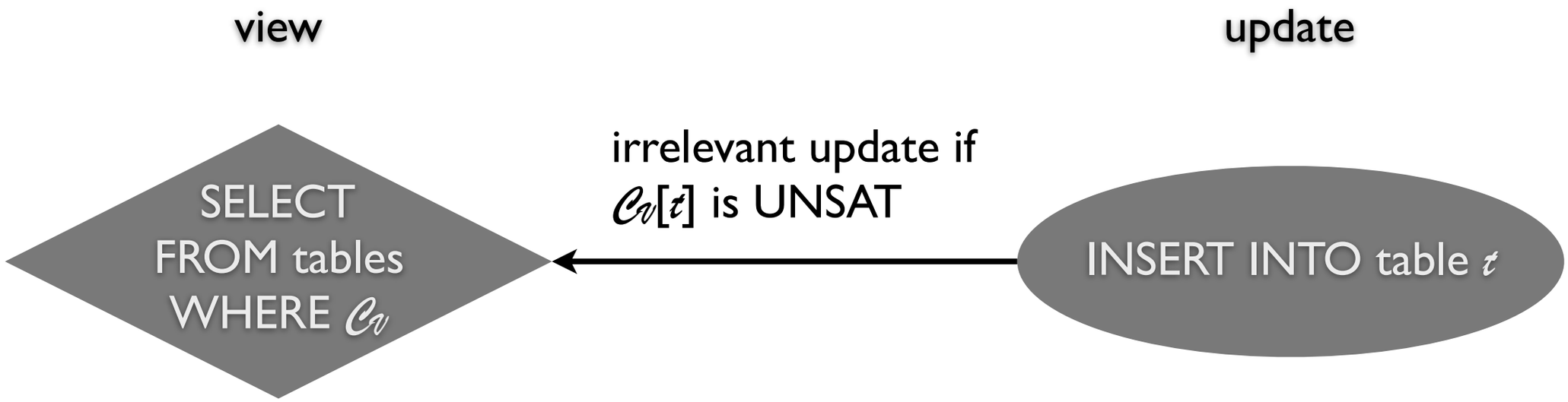
irrelevance reasoning for SDN



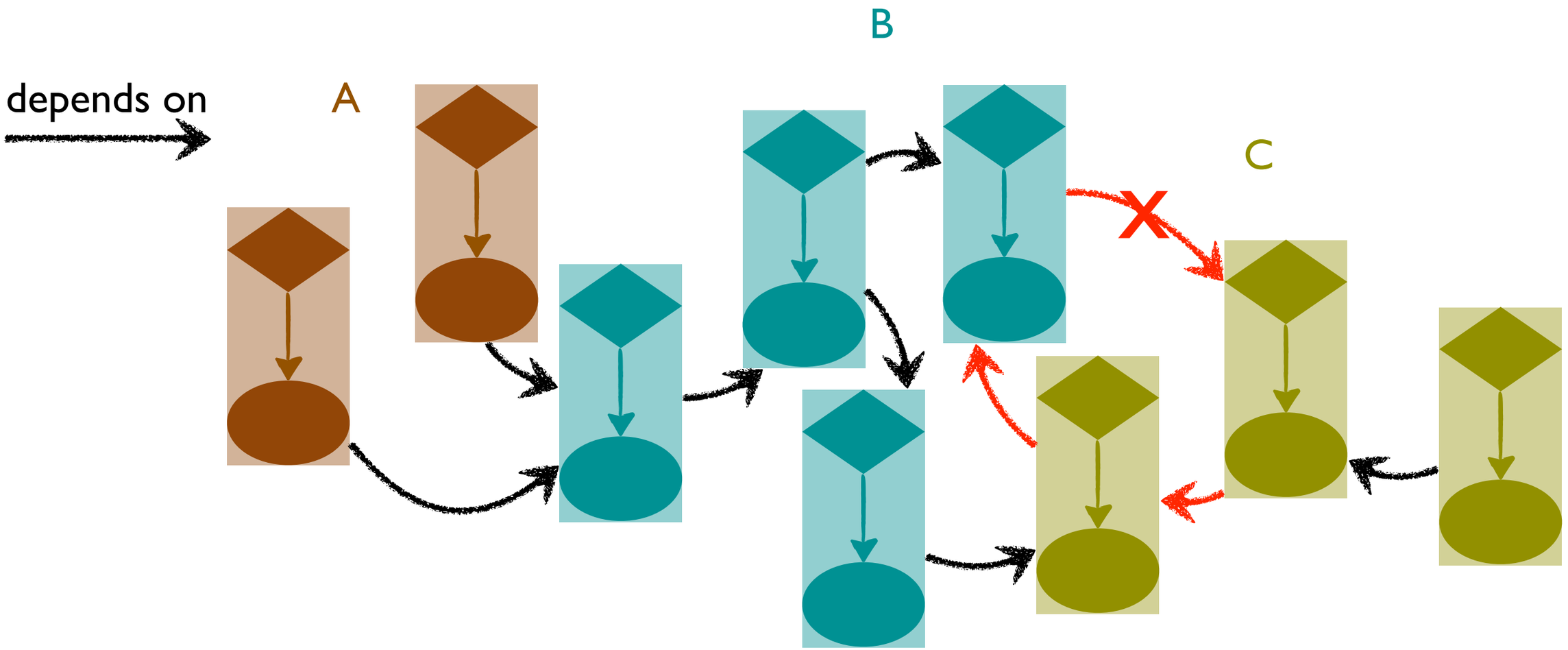
detect irrelevant database updates



detect irrelevant database update



using irrelevance reasoning for SDN



construct dependency graph

topological sort

- remove conflicts with user guidance
- assign each update a stratum number

synthesize a master orchestrator

- activate an update only when all updates with smaller stratum numbers have completed

using irrelevance reasoning for SDN

open questions

obtain the database representation

- use *Ravel*, a database-defined control platform
  - ravel-net.org

extract the database representation from arbitrary control software

- manually construct a map between data objects and database tables
- automatically convert data updates to DB write with conditions?
- extract view condition?

limitation

distribution and concurrency

- the network data plane is a distributed system with concurrent updates
- SDN relies on multiple controller for scalability

combine DB concurrency control and irrelevance reasoning?

thank you