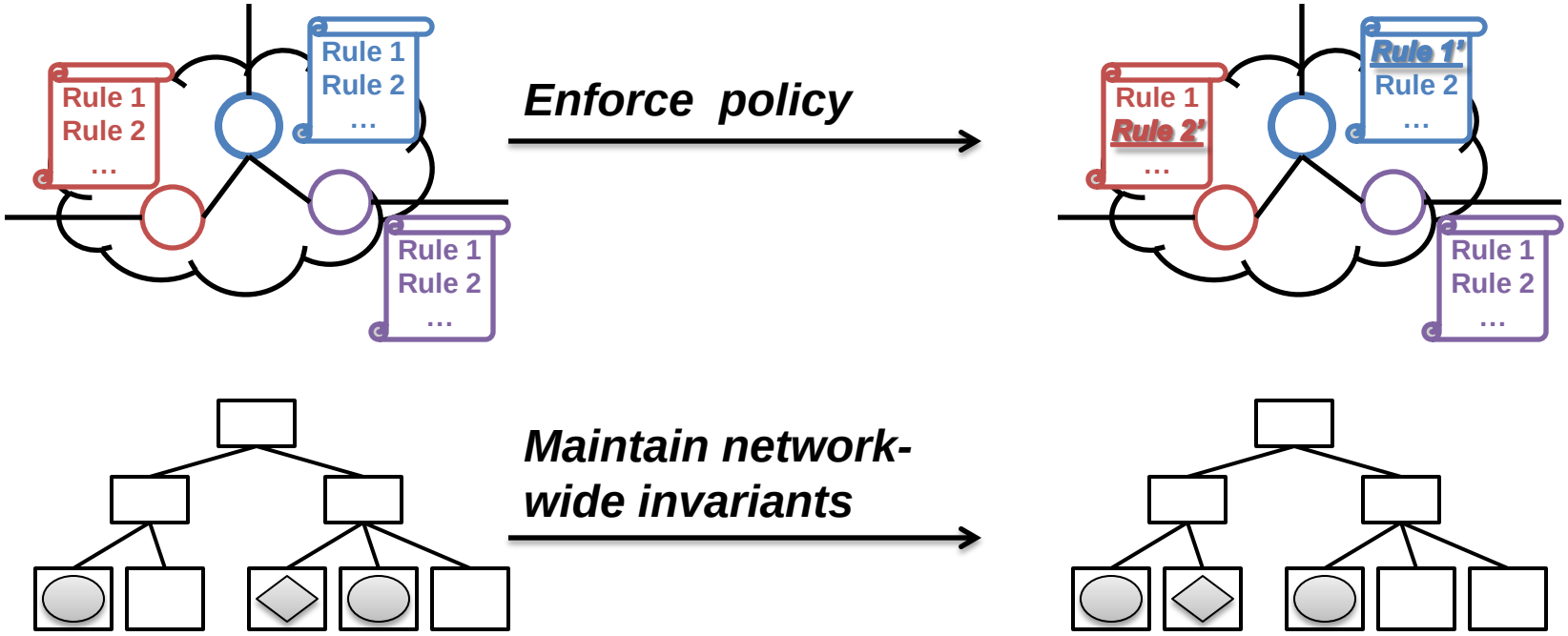
Formal Synthesis in Software-Defined Networks

Anduo Wang
University of Pennsylvania

A Project from the NSF Expeditions on Software Synthesis
http://excape.cis.upenn.edu

Management *Challenges*

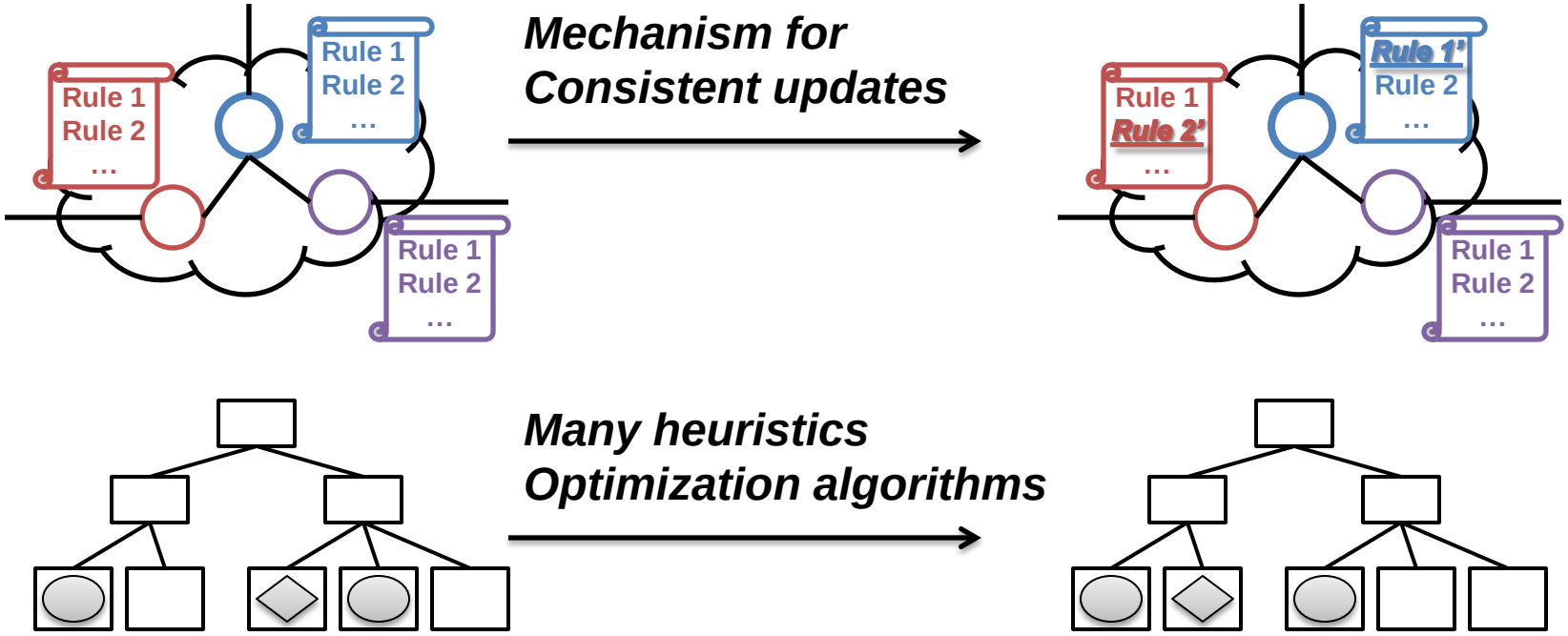
Network Migration



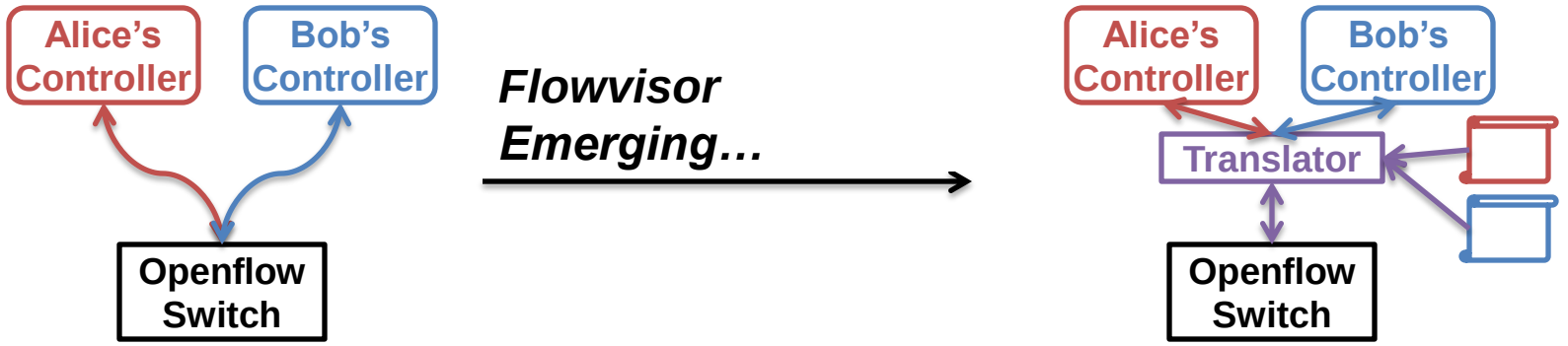
Network Virtualization

Management *Solutions*

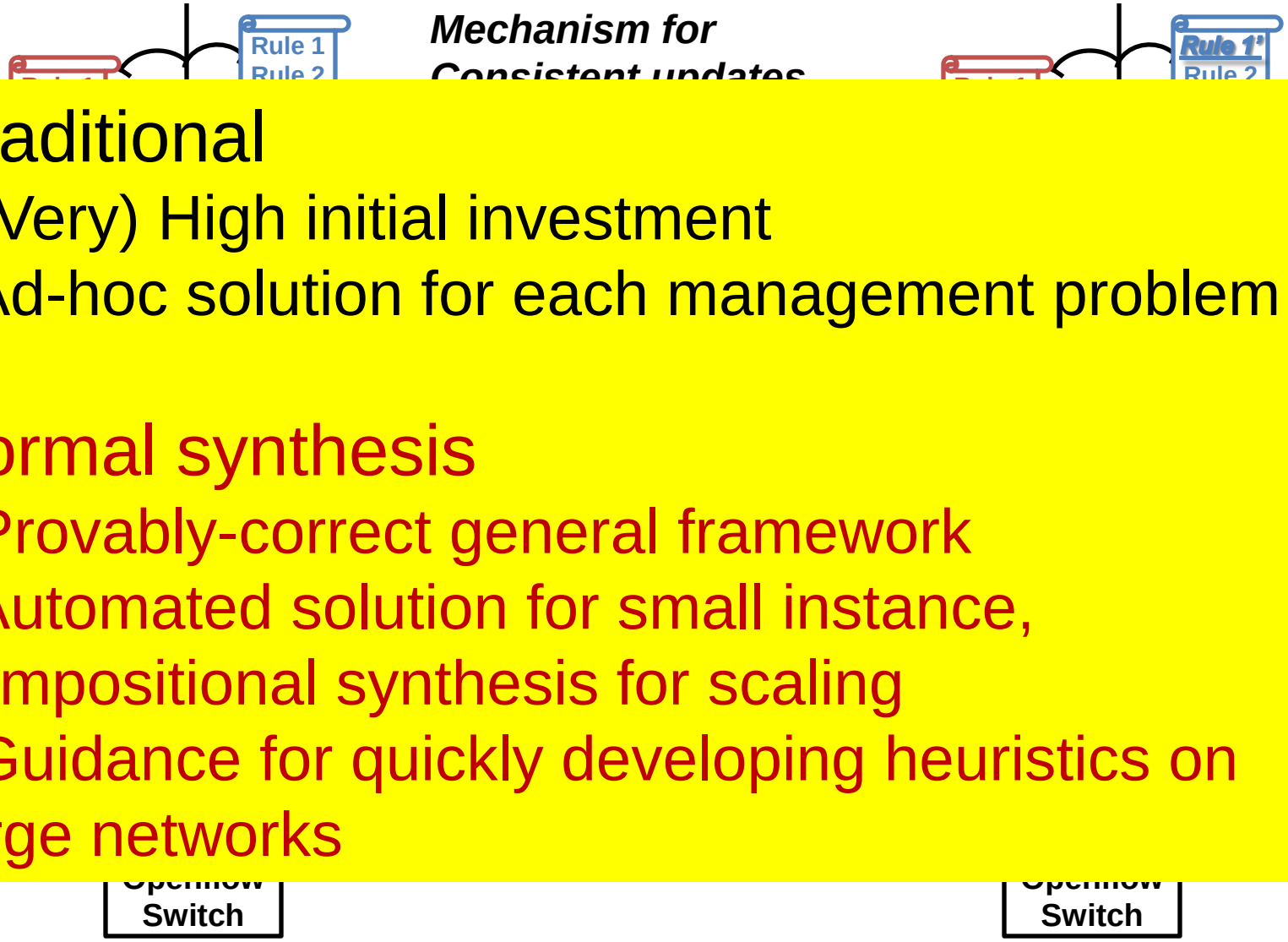
Network Migration



Network Virtualization



Formal Synthesis Approach



Traditional

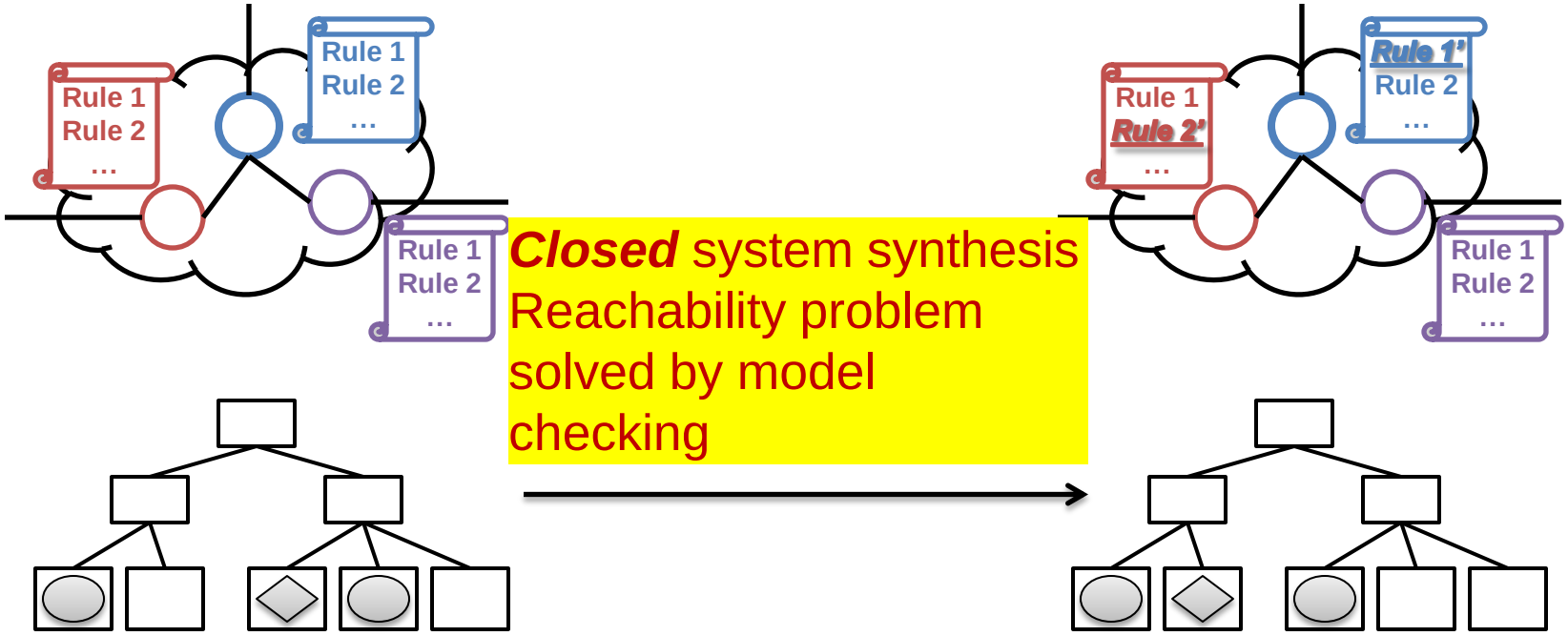
- (Very) High initial investment
- Ad-hoc solution for each management problem

Formal synthesis

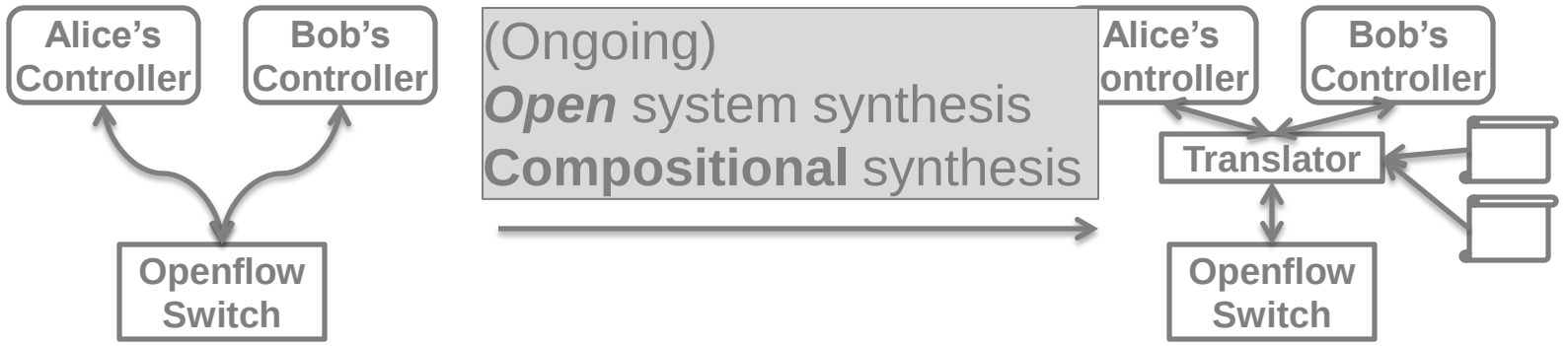
- Provably-correct general framework
- Automated solution for small instance, compositional synthesis for scaling
- Guidance for quickly developing heuristics on large networks

Formal Synthesis Approach

Network Migration



Network Virtualization



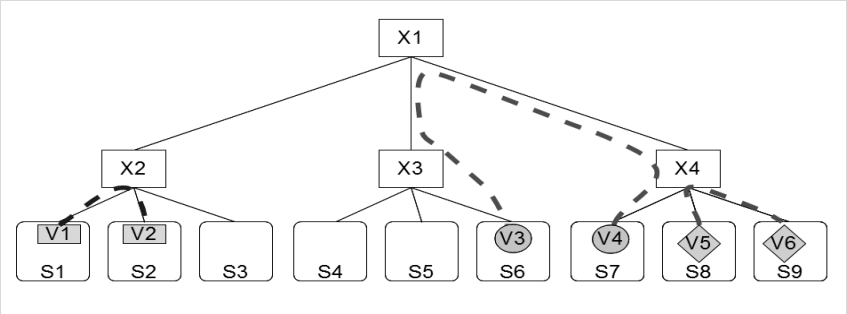
Closed System Synthesis for Migration

- Formulate network migration as reachability problem
  - Model network migration by a transition system
  - Find a migration ordering t (a sequence of atomic updates) from initial network state to the target final state s.t. constraints P holds during all transient states along t
  - A migration t exists if ¬P does not always hold
- Solving by model checker
  - Model check ¬P on the transition system
  - Counter example of ¬P gives t

Solution for VM Migration

```
-- specification !(( F ((v2 = s4 & v4 = s5) & v5 = s7) &  G ((v2 != v4 & v2 != v
5)  & v4 != v5)) &  G bandwidth13 < 2)  is false
-- as demonstrated by the followin
Trace Description: LTL Counterexam
Trace Type: Counterexample
-> State: 1.1 <-
  v2 = s2
  v4 = s7
  v5 = s8
  bandwidth13 = 1
-> State: 1.2 <-
  v4 = s5
-> State: 1.3 <-
  v2 = s4
-- Loop starts here
-> State: 1.4 <-
  v5 = s7
-- Loop starts here
```

- [HotSDN'12] Walk the Line: Consistent Network Updates with Bandwidth Guarantees

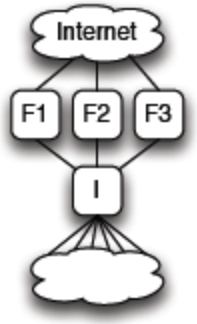


- Migration goal
  - Move V5, V4, V2 to S7, S5, S4.
- Constraints P
  - One substrate node can hold only one VM
  - Heavy dashed lines show inter-VM communications.
- Solution
  - Migrating with sequence V4, V2, V5 succeeds to migrate all nodes while migration with sequence V5, V2, V4 can migrate only one node

Solution for Configuration Migration

```
-- specification !( F ((I_g = F2 & I_s = F3) & F2_ssh = Monitor) &  G (((u_ssh_r
each = Deny & g_ssh_reach = Deny) & s_ssh_reach = Allow) & f_ssh_reach = Allow))
  is false
-- as demonstrated by the followi
Trace Description: LTL Counterexa
Trace Type: Counterexample
-> State: 1.1 <-
  I_u = F1
  I_g = F1
  I_s = F2
  I_f = F3
  F1_ssh = Monitor
  F2_ssh = Allow
  u_ssh_reach = Deny
  g_ssh_reach = Deny
  s_ssh_reach = Allow
  f_ssh_reach = Allow
-> State: 1.2 <-
  I_s = F3
-- Loop starts here
-> State: 1.3 <-
  F2_ssh = Monitor
-> State: 1.4 <-
  I_g = F2
-- Loop starts here
```

- [SIGCOMM'12] Abstractions for Network Update



Configuration I

	Type	Action
I	U, G	Forward F_1
	S	Forward F_2
	F	Forward F_3
F_1	SSH	Monitor
	$*$	Allow
F_2	$*$	Allow
F_3	$*$	Allow

→

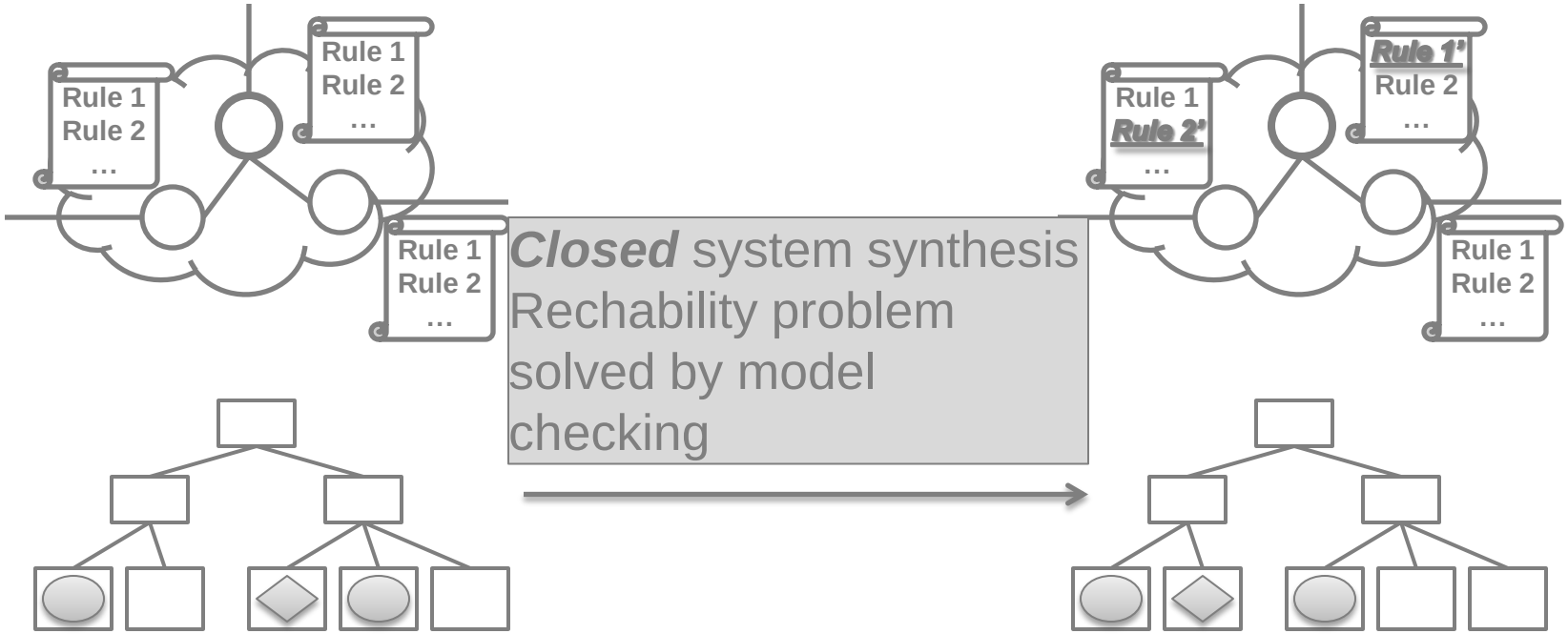
Configuration II

	Type	Action
I	U	Forward F_1
	G	Forward F_2
	S, F	Forward F_3
F_1	SSH	Monitor
	$*$	Allow
F_2	SSH	Monitor
	$*$	Allow
F_3	$*$	Allow

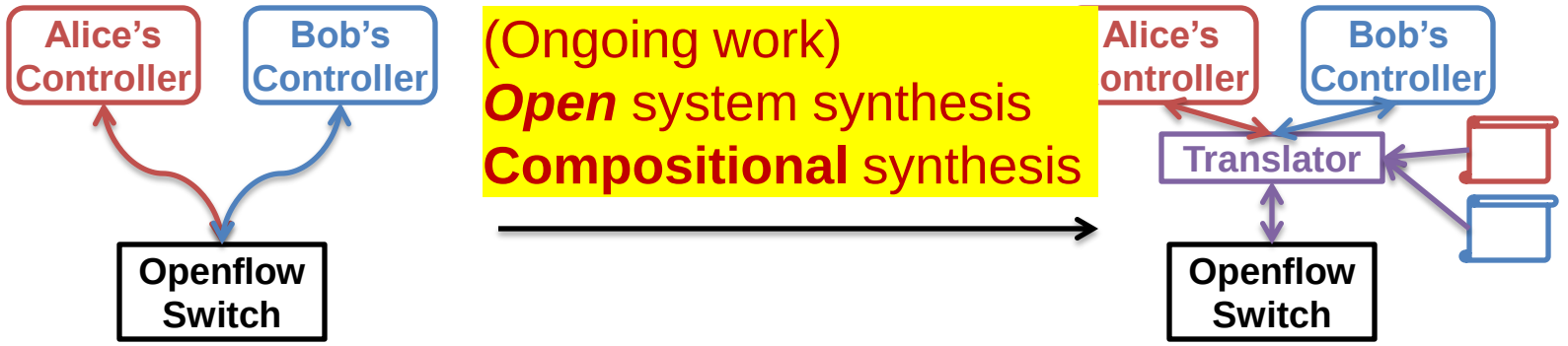
- Migration goal
  - Configuration I -> Configuration II
- Constraints P
  - Enforce a security policy that denies SSH traffic from untrustworthy hosts, but allows all other traffic to pass through the network unmodified
- Solution
  - Update I to forward S traffic to F3
  - Update F2 to deny SSH packets
  - Update I to forward G traffic to F2

Formal Synthesis Approach

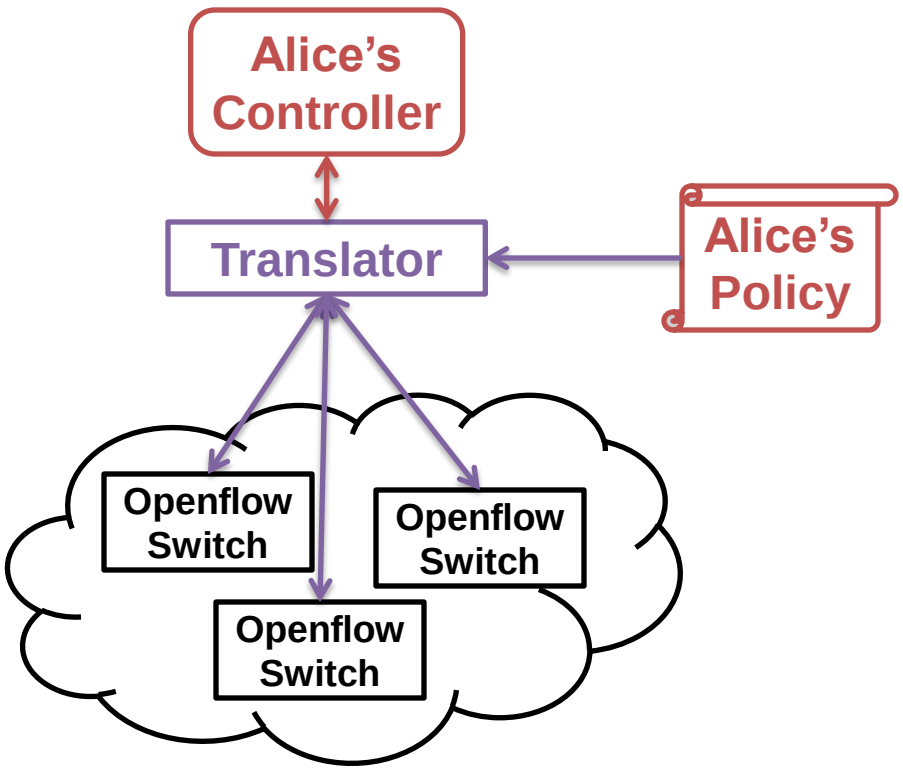
Network Migration

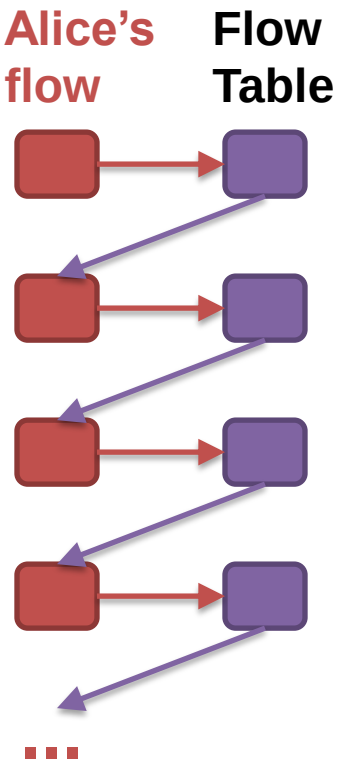


Network Virtualization

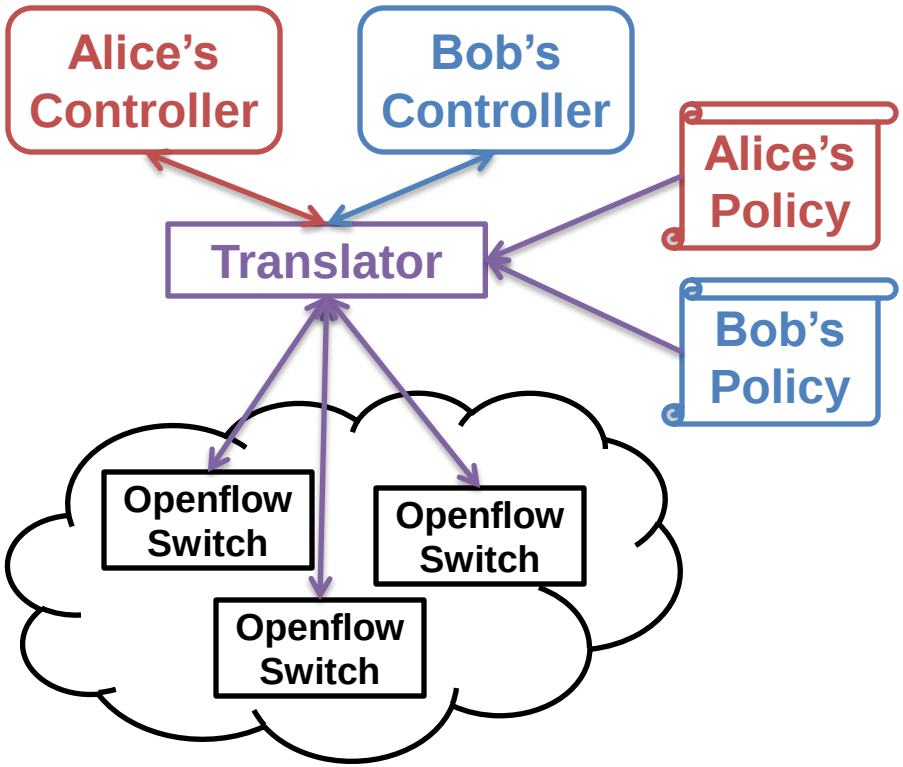


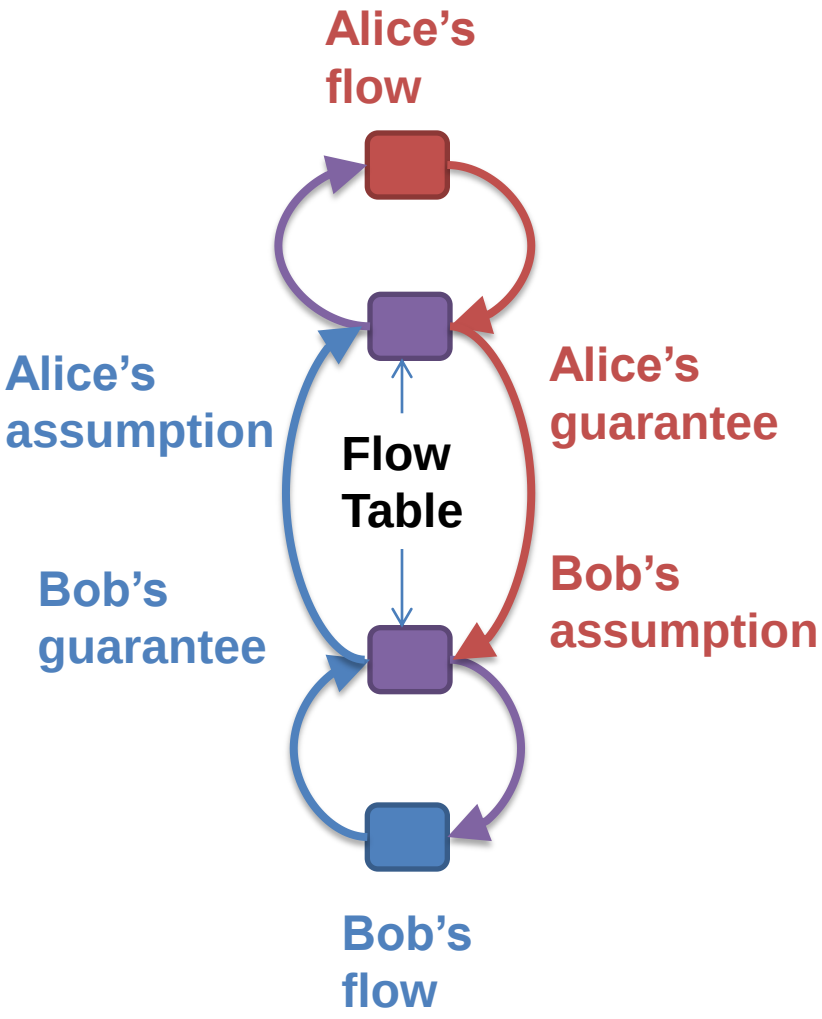
Open System Synthesis for Virtualization





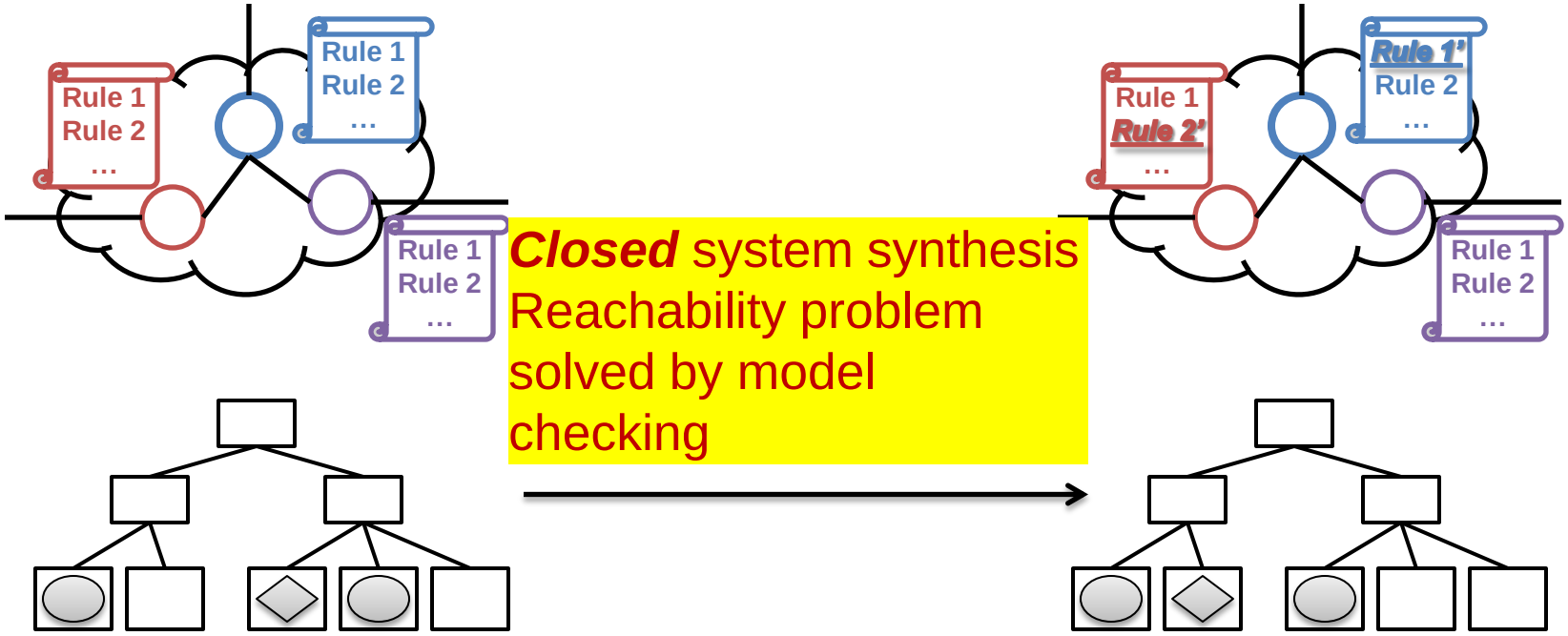
Compositional Synthesis for Virtualization





Conclusion: Formal Synthesis

Network Migration



Network Virtualization