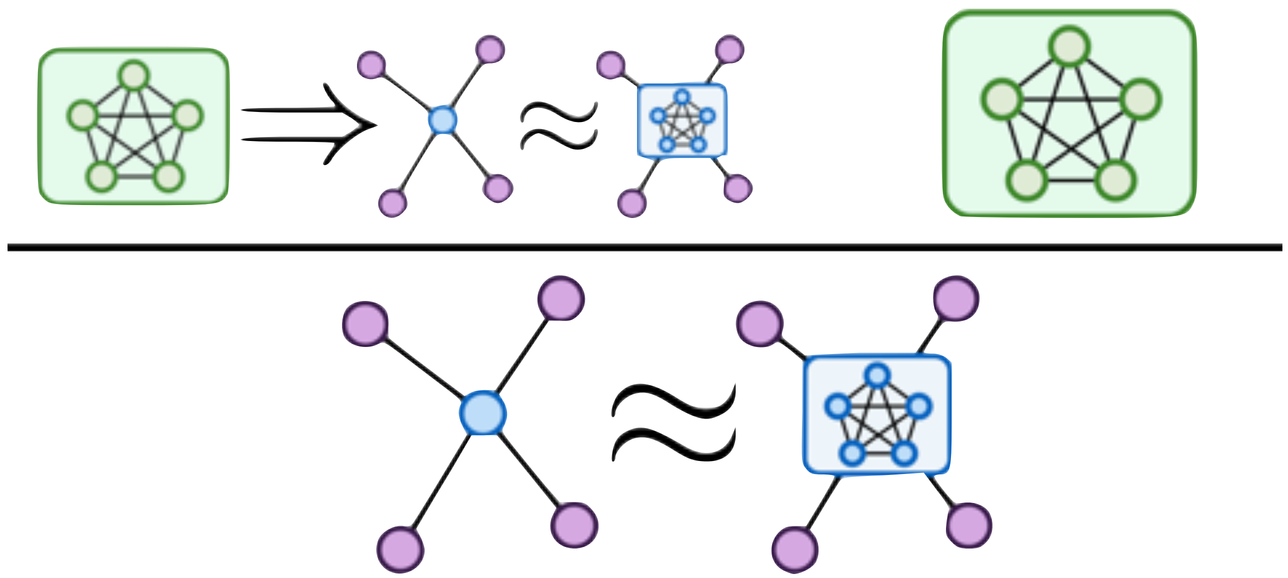
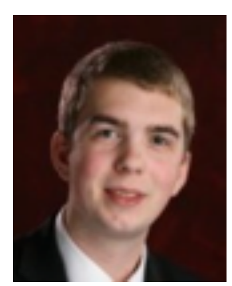
Planning for Change in a Formal Verification of the Raft Consensus Protocol



Doug Woos

James Wilcox



Steve Anton

Zach Tatlock

Mike Ernst

Tom Anderson

UNIVERSITY of WASHINGTON

Contributions

First formal proof of Raft's safety

first verified implementation!

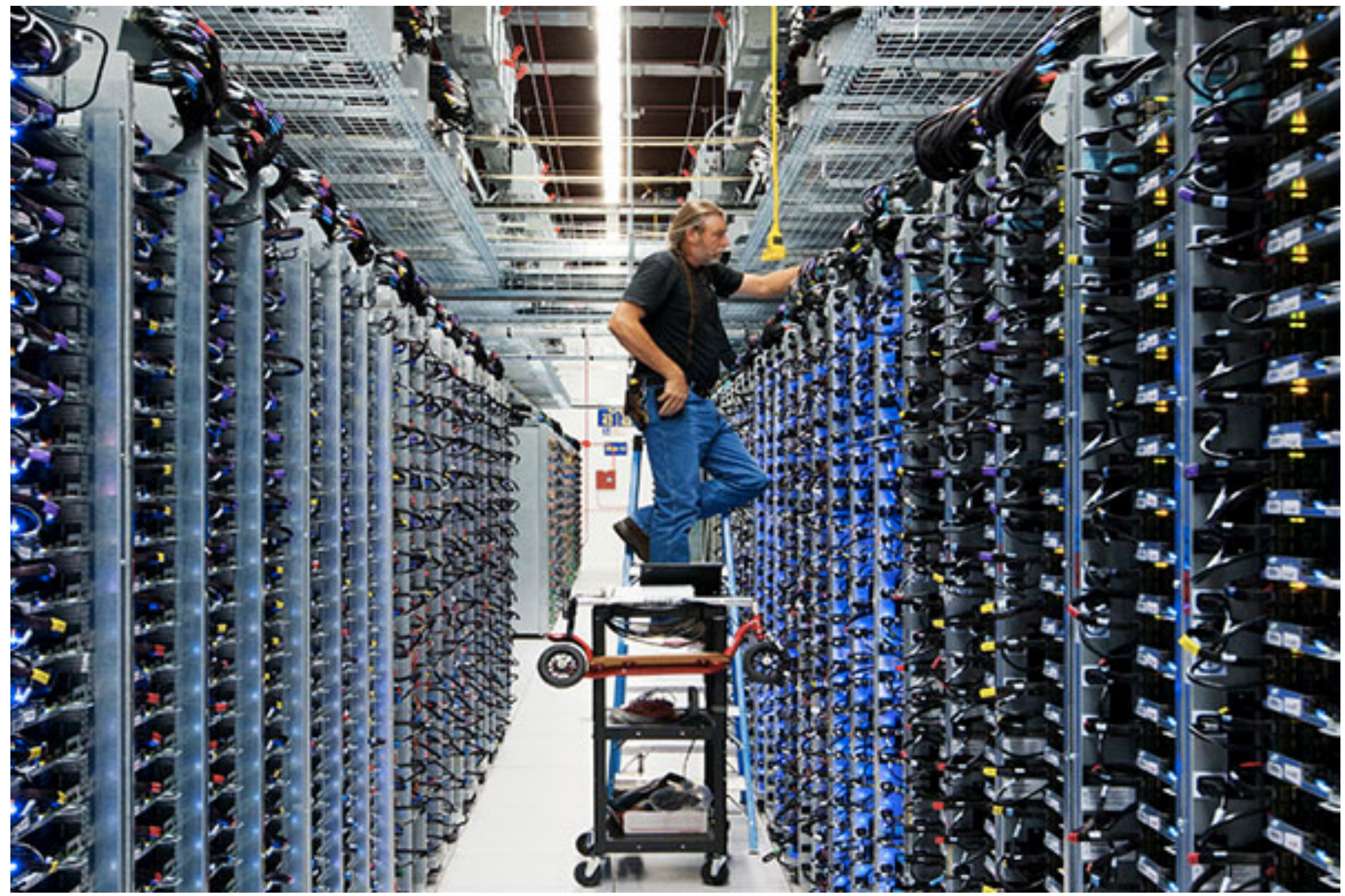
Large-scale Verdi case study

stress test; reverification inevitable

Proof engineering lessons

affinity lemmas, etc.

Distributed Systems



Reliably deliver procrastination

Also serious infrastructure

One day last summer...

The New York Times

The Stock Market Bell Rings, Computers Fail, Wall Street Cringes

By NATHANIEL POPPER JULY 8, 2015

Problems with technology have at times roiled global financial markets, but the 223-year-old New York Stock Exchange has held itself up as an oasis of humans ready to step in when the computers go haywire.

On Wednesday, however, those working on the trading floor were left helpless when the computer systems at the exchange went down for nearly four hours in the middle of the day, bringing an icon of capitalism's ceaseless energy to a costly halt.

The exchange ultimately returned to action shortly before the closing bell,

One day last summer...

The New York Times

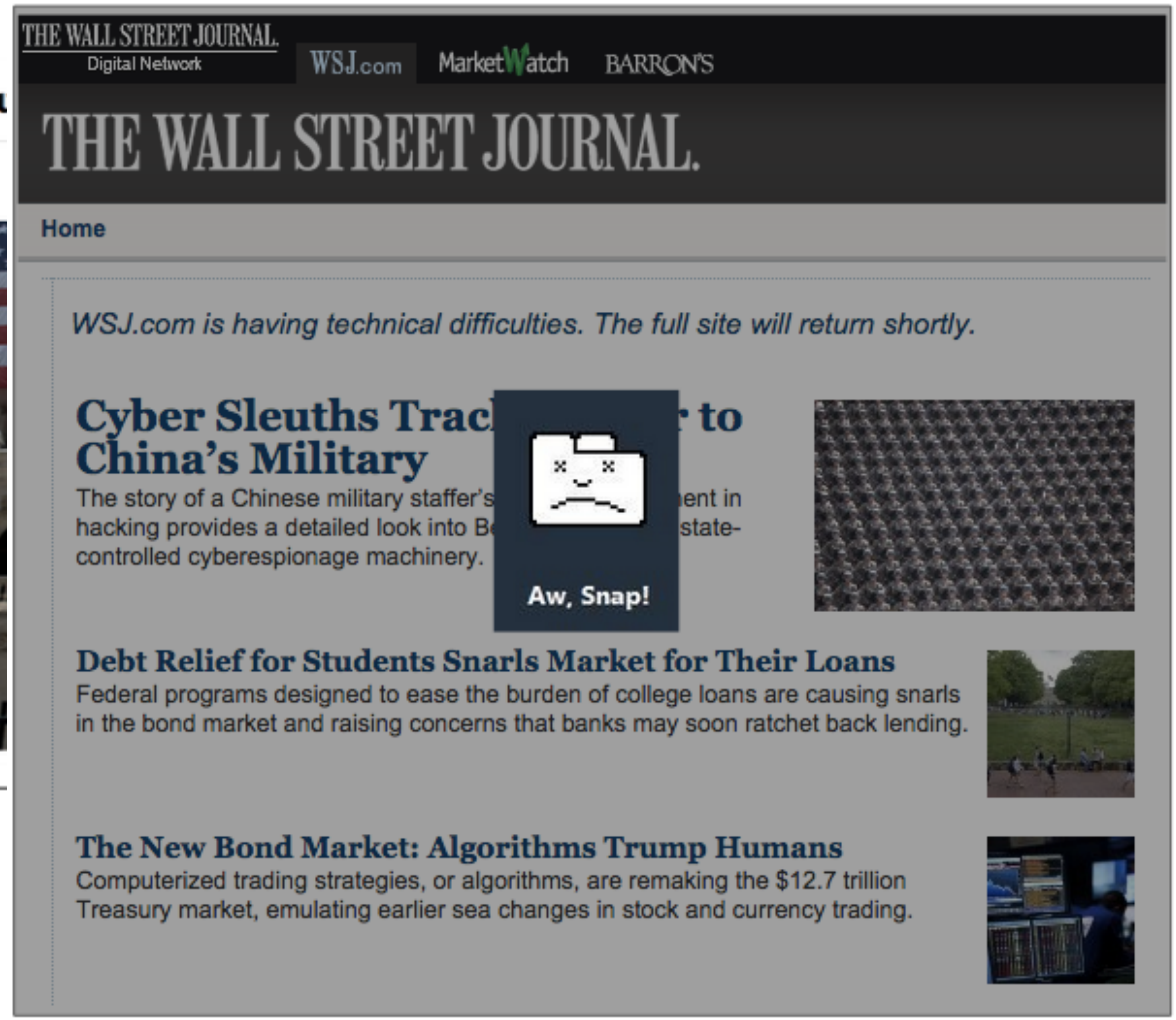
The Stock Market Bell Rings, Compu

By NATHANIEL POPPER JULY 8, 2015

Problems with technology have at times roiled global financial markets, but the 223-year-old New York Stock Exchange has held itself up as an oasis of humans ready to step in when the computers go haywire.

On Wednesday, however, those working on the trading floor were left helpless when the computer systems at the exchange went down for nearly four hours in the middle of the day, bringing an icon of capitalism's ceaseless energy to a costly halt.

The exchange ultimately returned to action shortly before the closing bell,



THE WALL STREET JOURNAL. Digital Network | WSJ.com | MarketWatch | BARRON'S

THE WALL STREET JOURNAL.

Home

WSJ.com is having technical difficulties. The full site will return shortly.

Cyber Sleuths Trac… r to China's Military

The story of a Chinese military staffer's … ent in hacking provides a detailed look into Be… state-controlled cyberespionage machinery.

Aw, Snap!

Debt Relief for Students Snarls Market for Their Loans

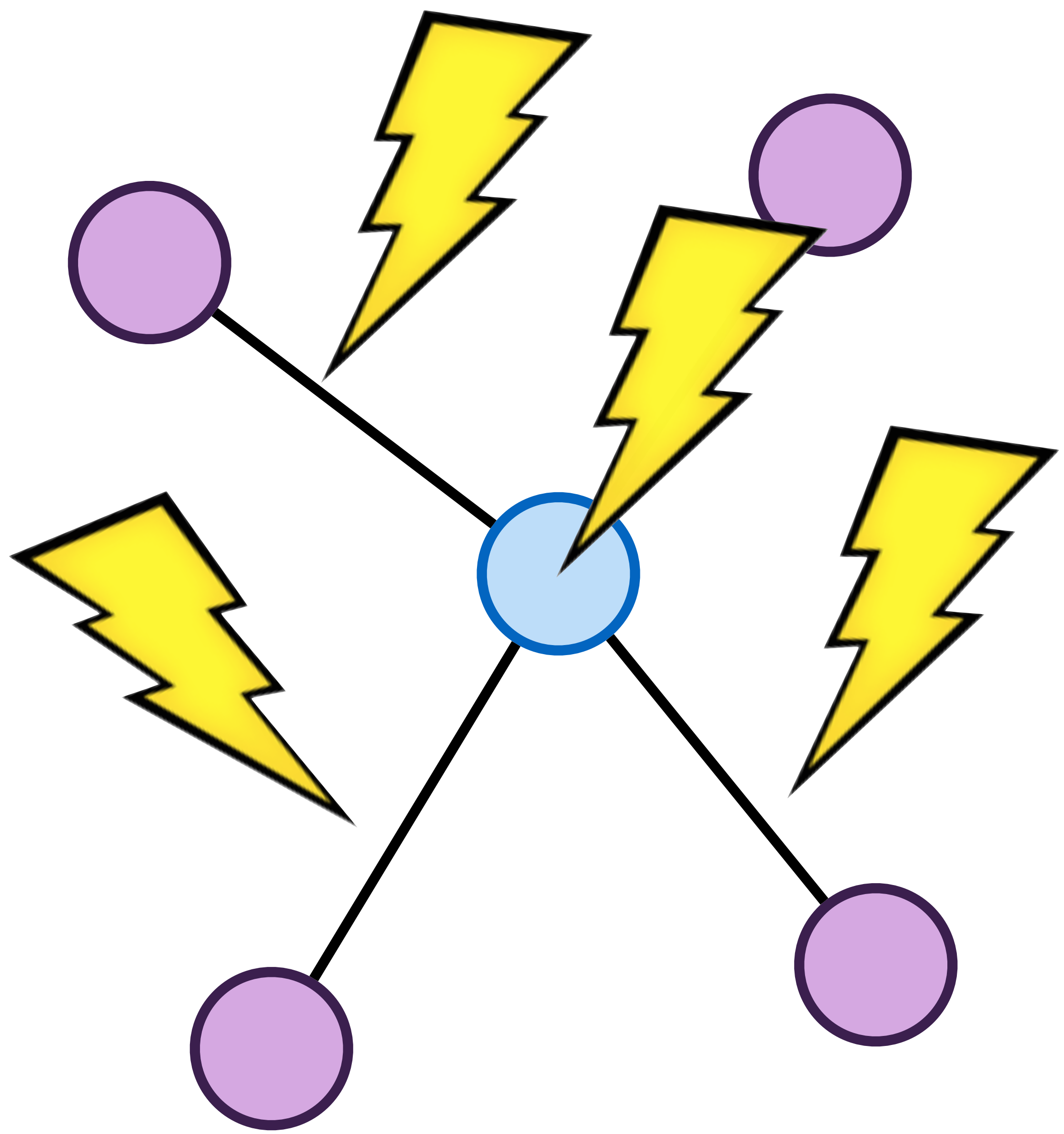
Federal programs designed to ease the burden of college loans are causing snarls in the bond market and raising concerns that banks may soon ratchet back lending.

The New Bond Market: Algorithms Trump Humans

Computerized trading strategies, or algorithms, are remaking the $12.7 trillion Treasury market, emulating earlier sea changes in stock and currency trading.

One day last summer...

How distributed systems fail



Related Work

EventML [LADA12, AVoCS15]

language for verified distributed systems

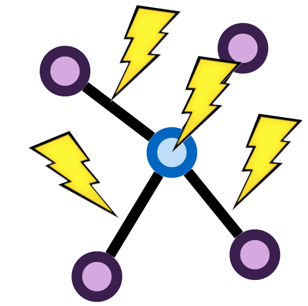
IronFleet [SOSP15]

liveness, log compaction, serialization

Verdi [PLDI15]

network semantics, transformers, higher-order

Verdi background

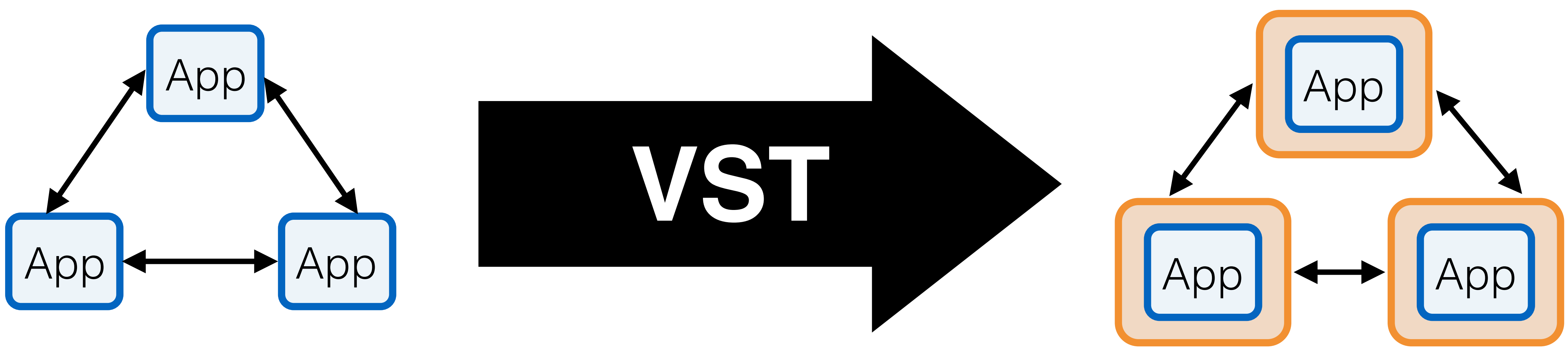


Network semantics

operational semantics define network behavior

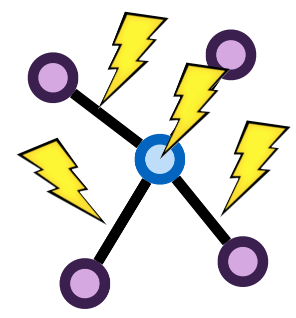
Verified system transformers

prove property transfer to adversarial network



$$\forall\ \Phi\ S,\ \texttt{holds}(\Phi,\ S,\ \rightsquigarrow_1) \rightarrow$$
$$\texttt{holds}(\texttt{transfer}(\Phi),\ T(S),\ \rightsquigarrow_2)$$

Big Picture



Past: Verdi Framework

compositional fault tolerance

Present: Verified Raft

critical piece of infrastructure

Future:

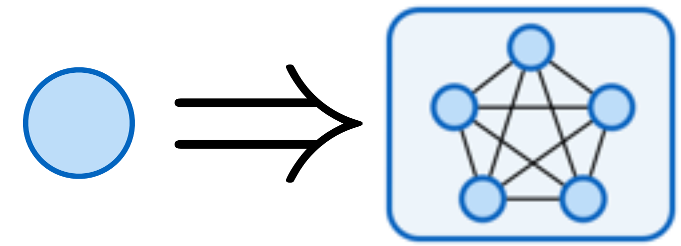
dynamically upgrading systems

program logic

Outline

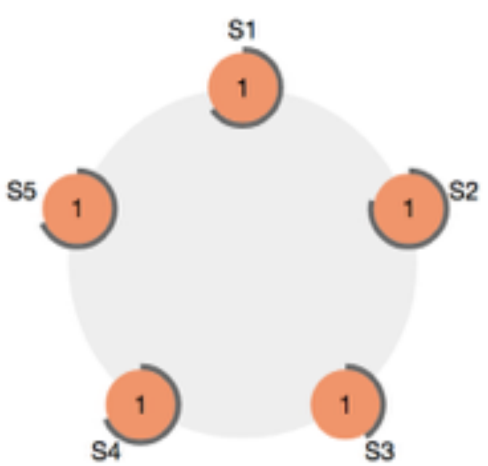
Verification Challenge

state machine replication



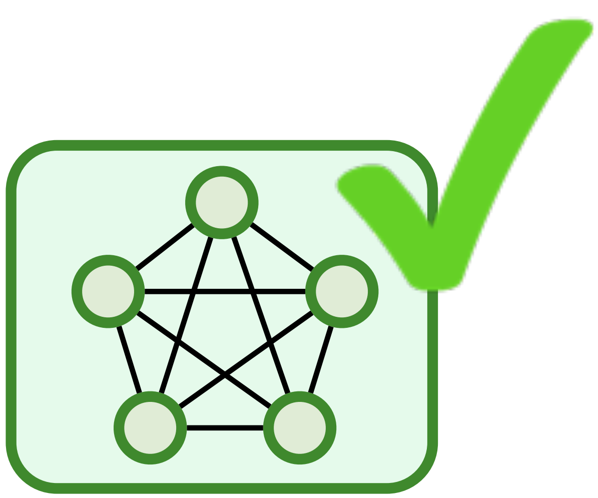
Raft Algorithm

implemented in Verdi

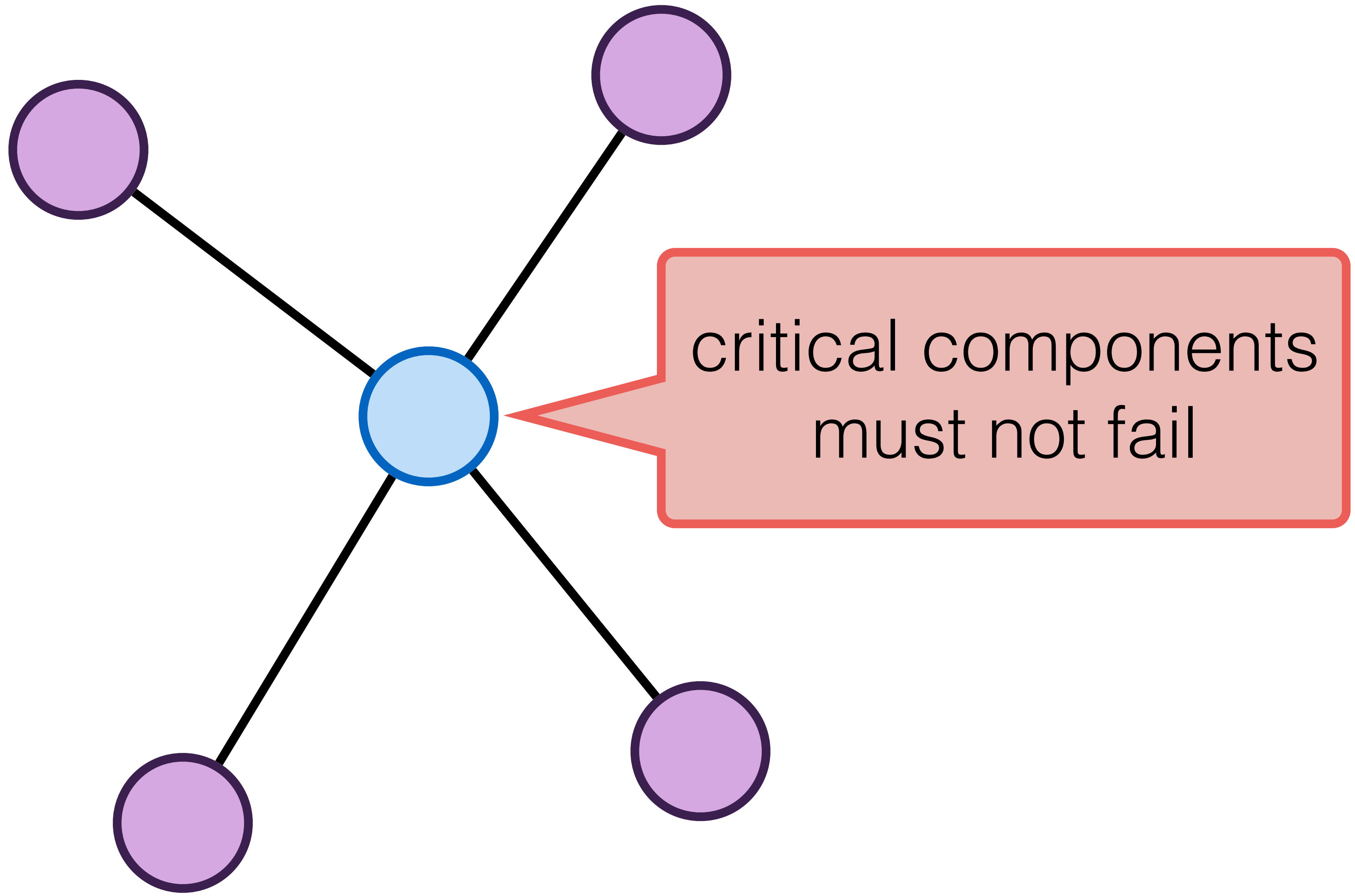


Proof Overview

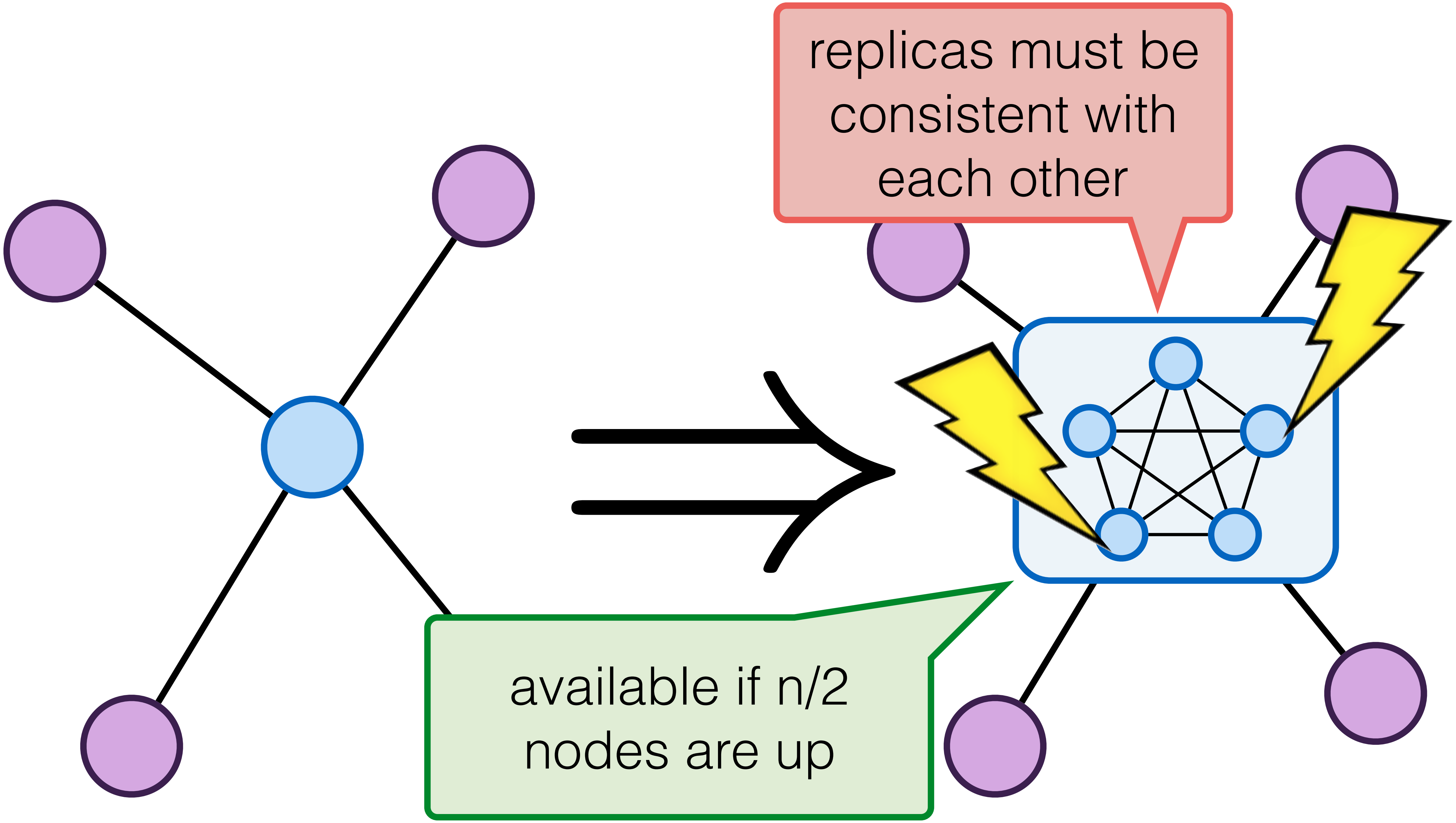
and lessons learned



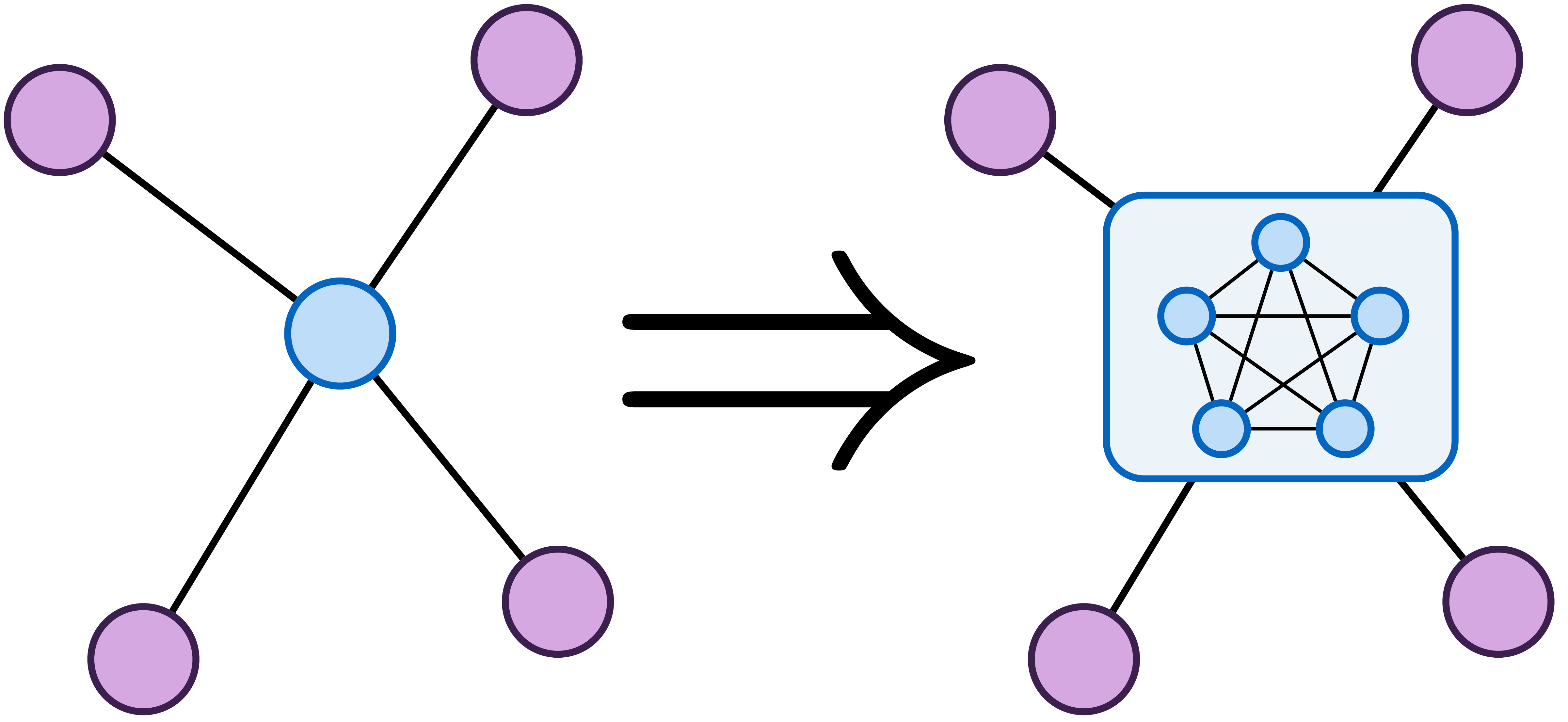
Replication for fault tolerance



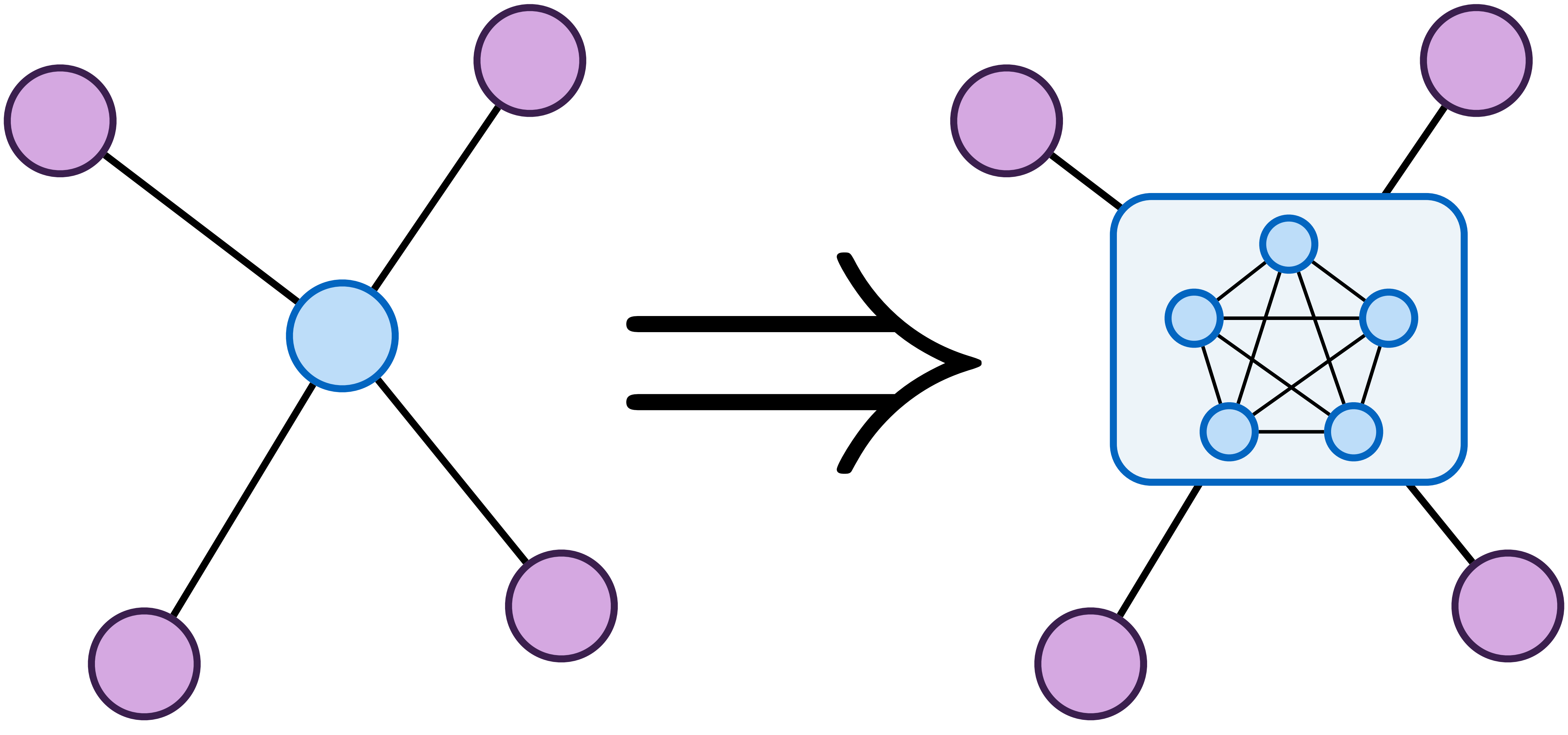
Replication for fault tolerance



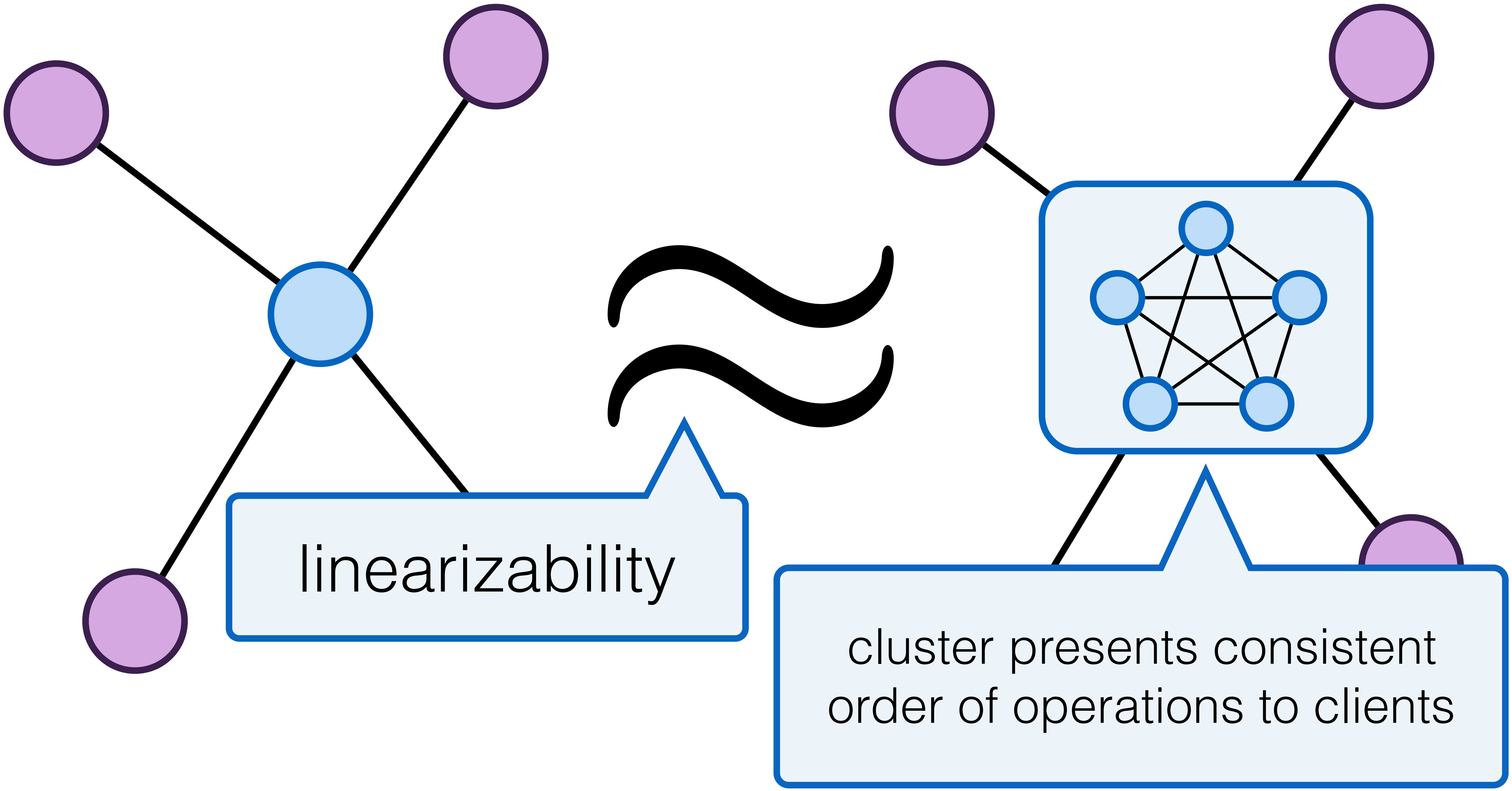
Replication for fault tolerance



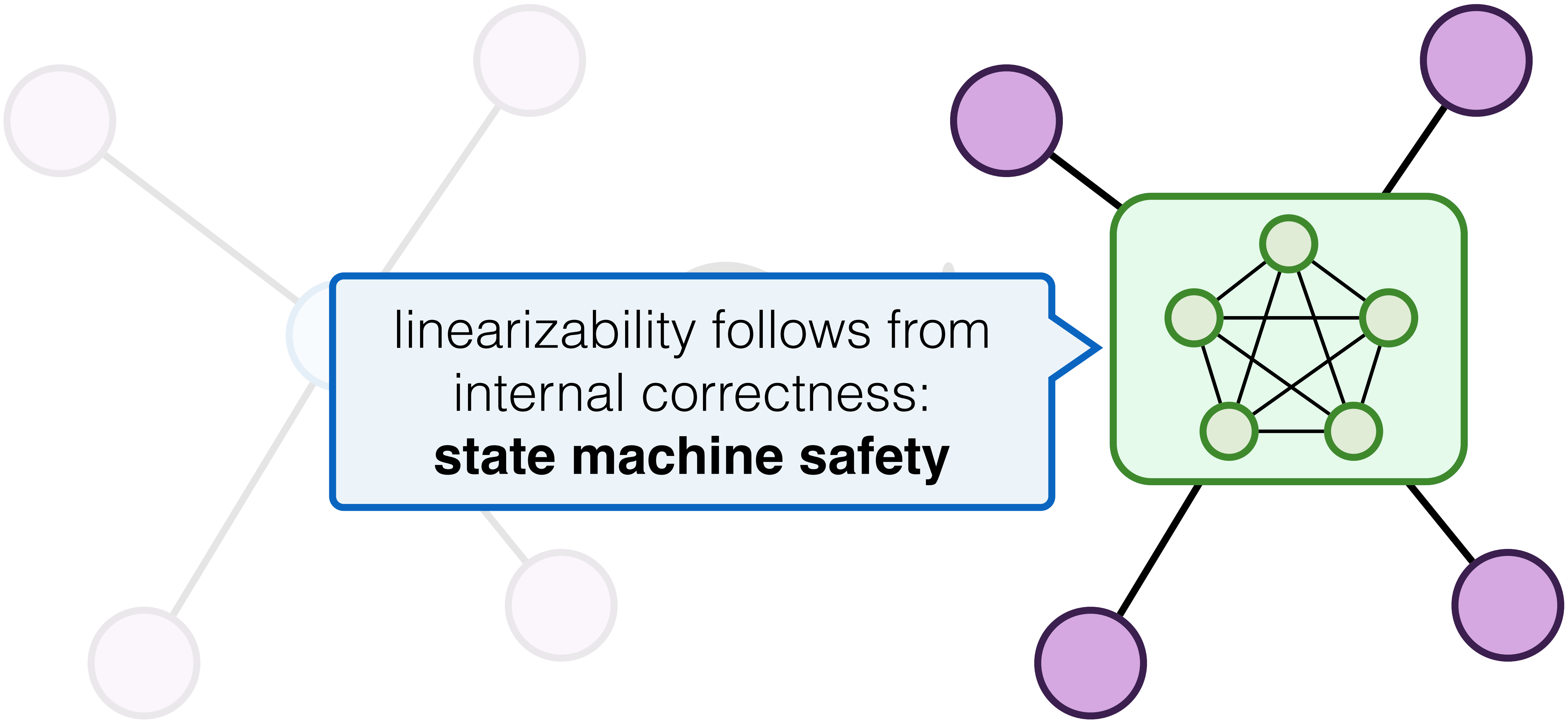
Replication correctness



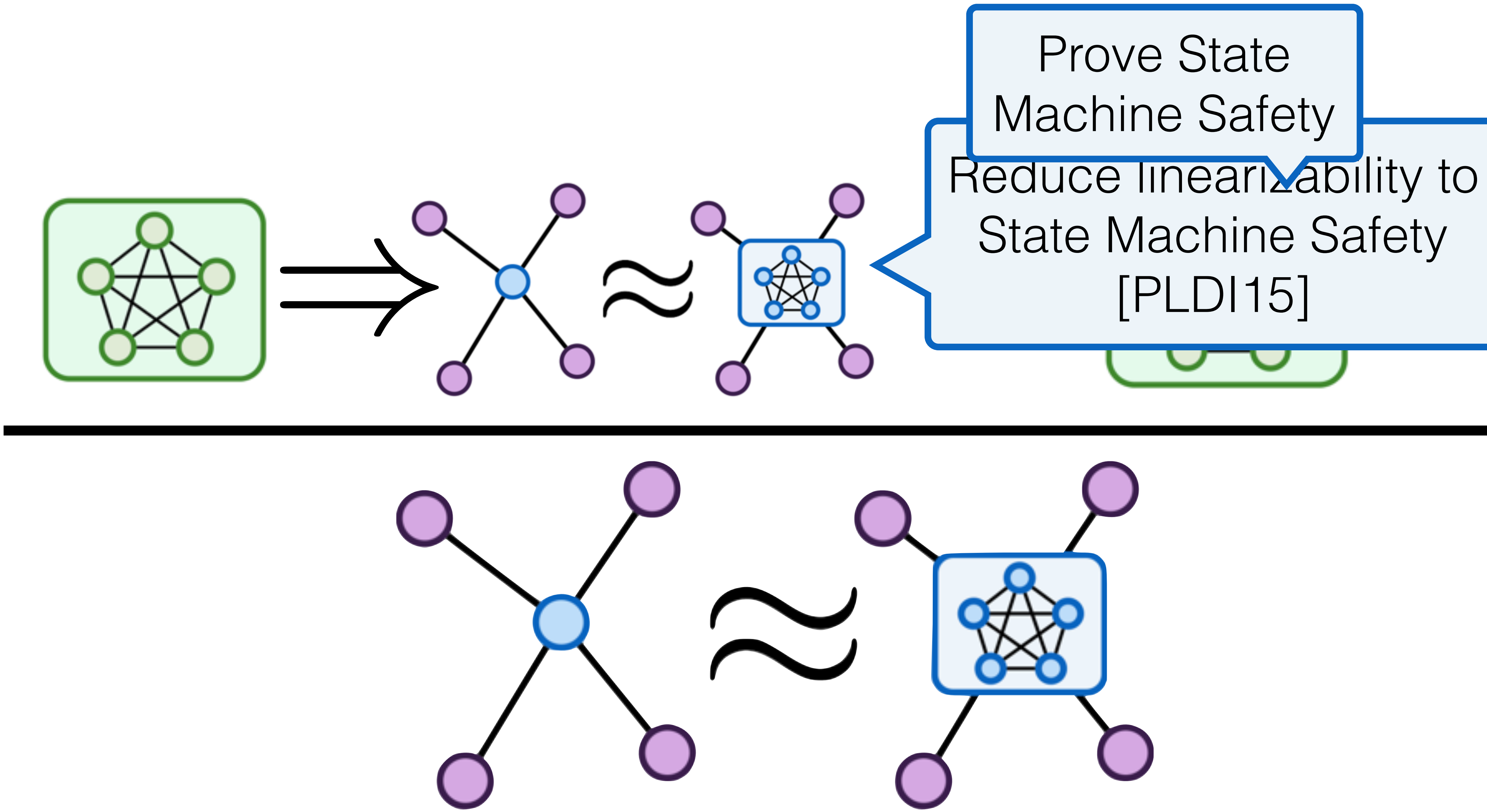
Replication correctness



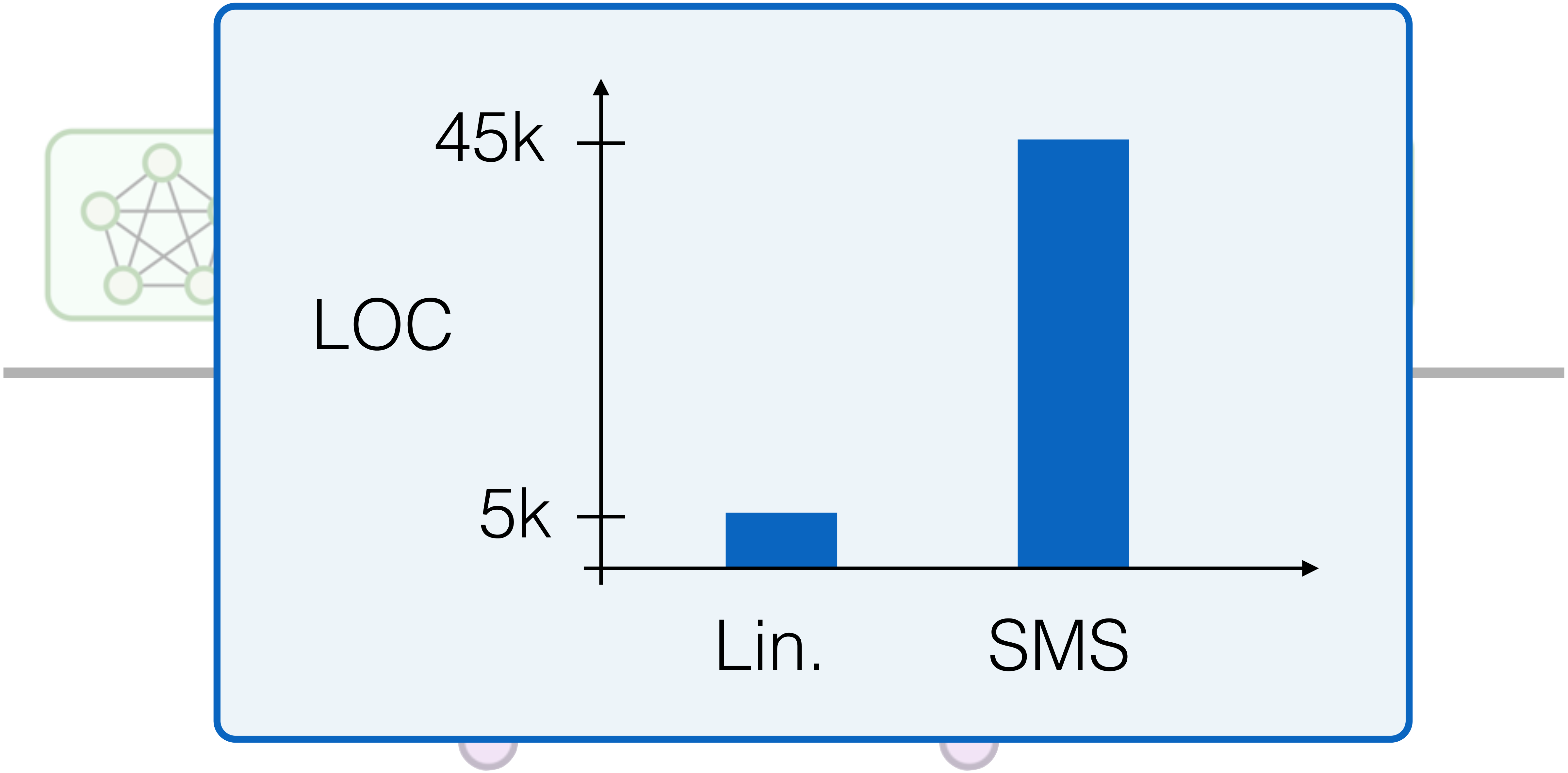
Internal Correctness



Goal: Verify Raft



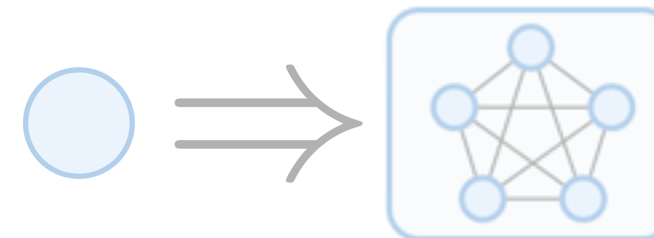
Goal: Verify Raft



Outline

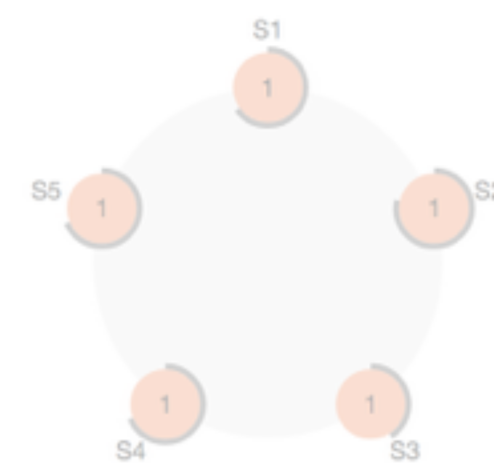
Verification Challenge

state machine replication



Raft Algorithm

implemented in Verdi



Proof Overview

and lessons learned

Formalizing the network

state of the world

$$(P, \Sigma, T)$$

packets in flight

history of I/O

data @ nodes

Formalizing the network

$$(P, \Sigma, T)$$

Formalizing the network

$$(P, \Sigma, T) \rightsquigarrow (P', \Sigma', T')$$

Defining network semantics

$$\frac{H_{\text{net}}(dst,\ \Sigma[dst],\ src,\ m) = (\sigma',\ o,\ P') \qquad \Sigma' = \Sigma[dst \mapsto \sigma']}{(\{(src,\ dst,\ m)\} \uplus P,\ \Sigma,\ T) \rightsquigarrow (P \uplus P',\ \Sigma',\ T \mathbin{++} \langle o \rangle)}\ \textsc{Deliver}$$

Defining network semantics

$$\frac{H_{\text{net}}(dst,\ \Sigma[dst],\ src,\ m) = (\sigma',\ o,\ P') \qquad \Sigma' = \Sigma[dst \mapsto \sigma']}{(\{(src,\ dst,\ m)\} \uplus P,\ \Sigma,\ T) \rightsquigarrow (P \uplus P',\ \Sigma',\ T \mathbin{++} \langle o \rangle)}\ \textsc{Deliver}$$

$$\frac{p \in P}{(P,\ \Sigma,\ T) \rightsquigarrow (P \uplus \{p\},\ \Sigma,\ T)}\ \textsc{Duplicate}$$

$$\frac{}{(\{p\} \uplus P,\ \Sigma,\ T) \rightsquigarrow (P,\ \Sigma,\ T)}\ \textsc{Drop}$$

$$\frac{H_{\text{tmt}}(n,\ \Sigma[n]) = (\sigma',\ o,\ P') \qquad \Sigma' = \ \Sigma[n \mapsto \sigma']}{(P,\ \Sigma,\ T) \rightsquigarrow (P \uplus P',\ \Sigma',\ T \mathbin{++} \langle \text{tmt},\ o \rangle)}\ \textsc{Timeout}$$

Defining network semantics

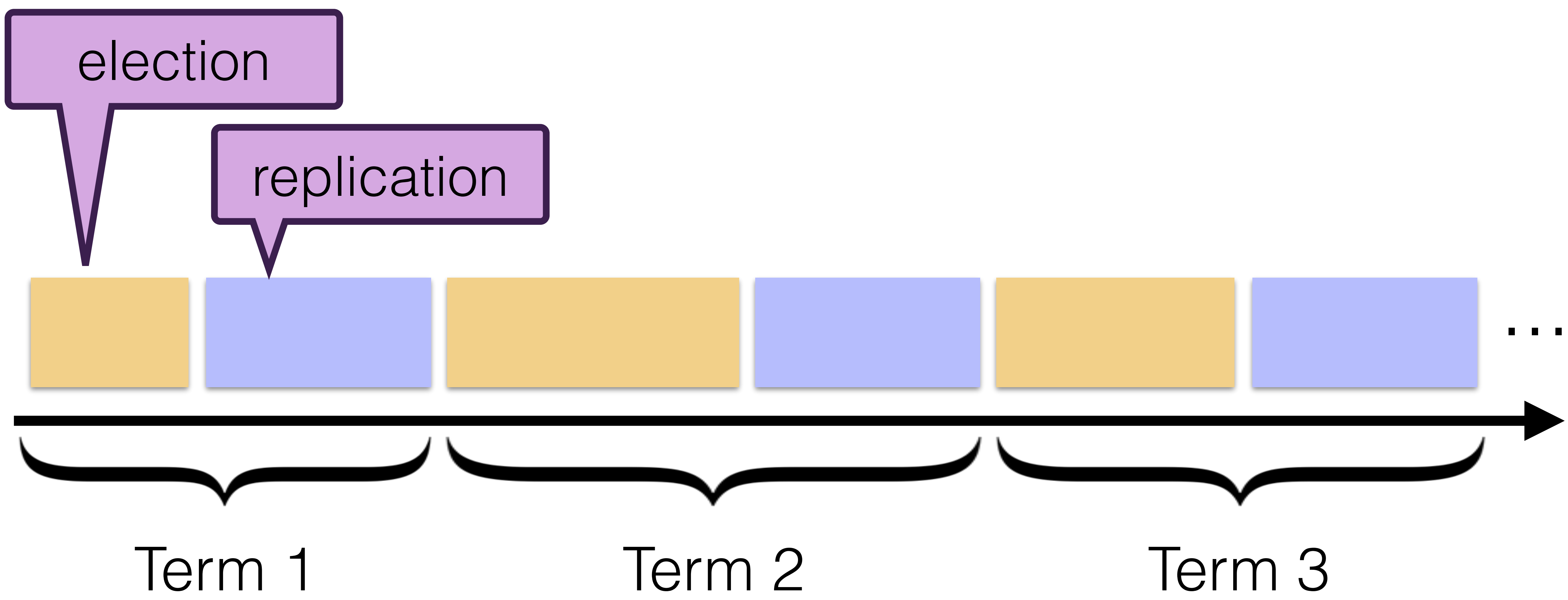
$$\frac{H_{\mathrm{net}}(dst,\ \Sigma[dst],\ src,\ m) = (\sigma',\ o,\ P') \qquad \Sigma' = \Sigma[dst \mapsto \sigma']}{(\{(src,\ dst,\ m)\} \uplus P,\ \Sigma,\ T) \rightsquigarrow (P \uplus P',\ \Sigma',\ T \mathbin{++} \langle o \rangle)}\ \textsc{Deliver}$$

systems defined by handlers

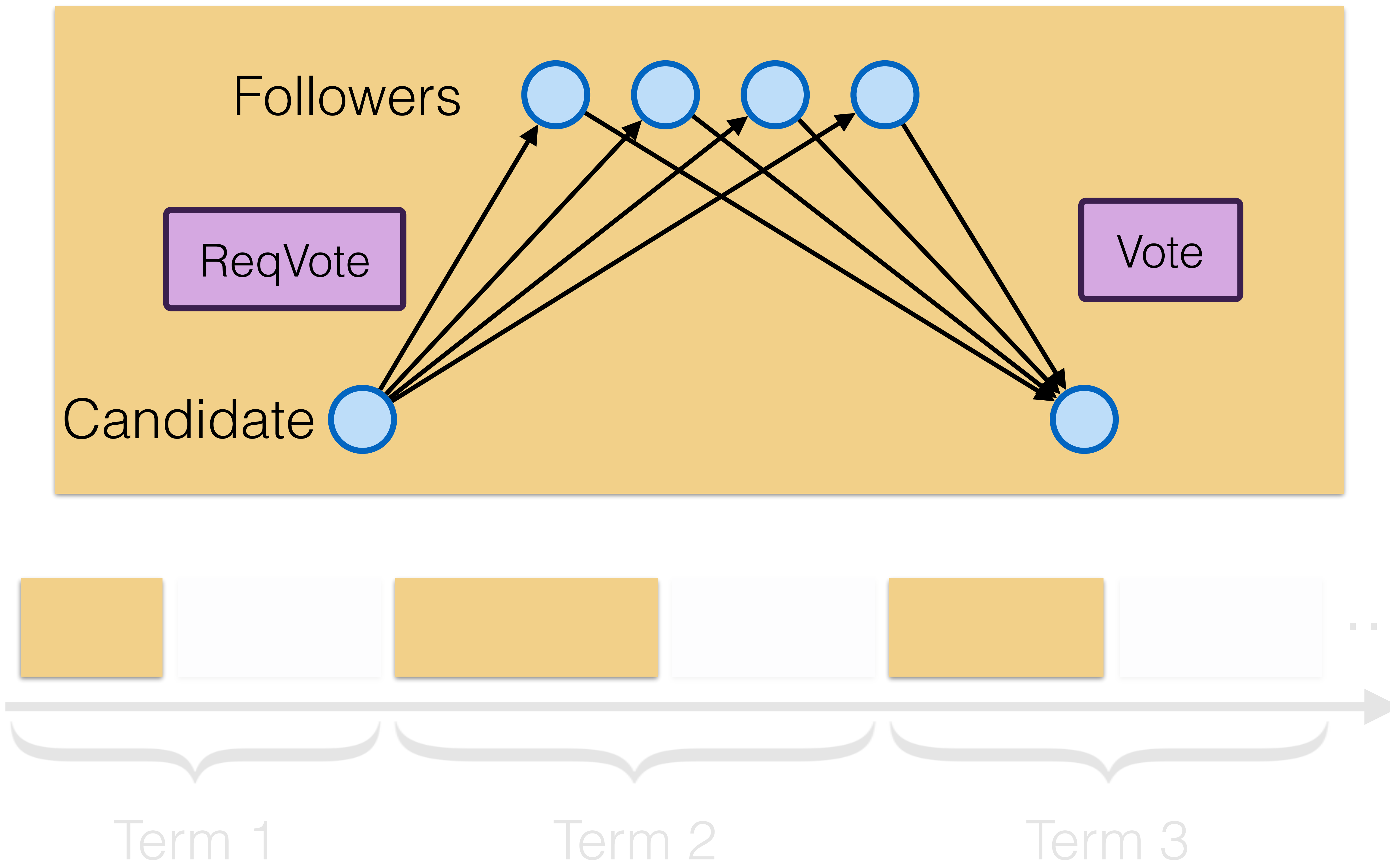
$$\frac{}{(\{p\} \uplus P,\ \Sigma,\ T) \rightsquigarrow (P,\ \Sigma,\ T)}\ \textsc{Drop}$$

$$\frac{H_{\mathrm{tmt}}(n,\ \Sigma[n]) = (\sigma',\ o,\ P') \qquad \Sigma' = \Sigma[n \mapsto \sigma']}{(P,\ \Sigma,\ T) \rightsquigarrow (P \uplus P',\ \Sigma',\ T \mathbin{++} \langle \mathrm{tmt},\ o \rangle)}\ \textsc{Timeout}$$

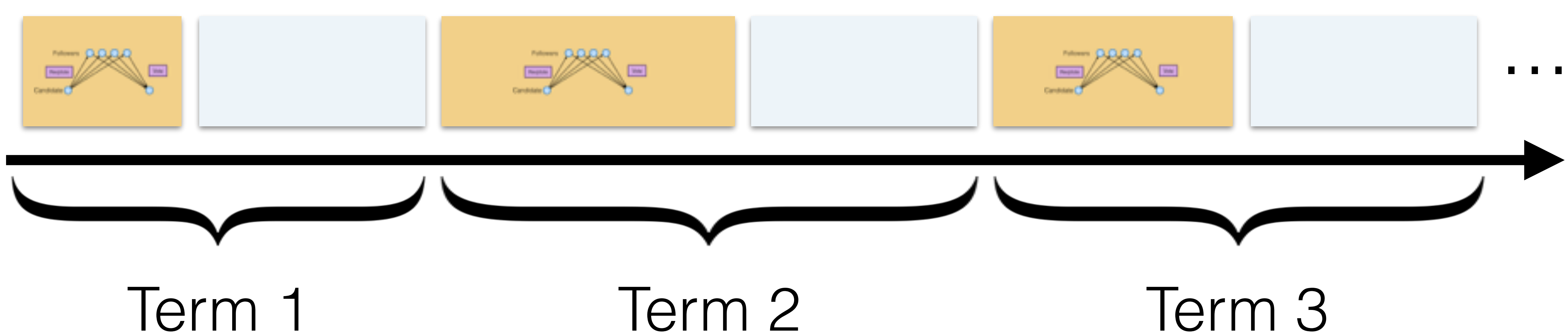
Implementing Raft



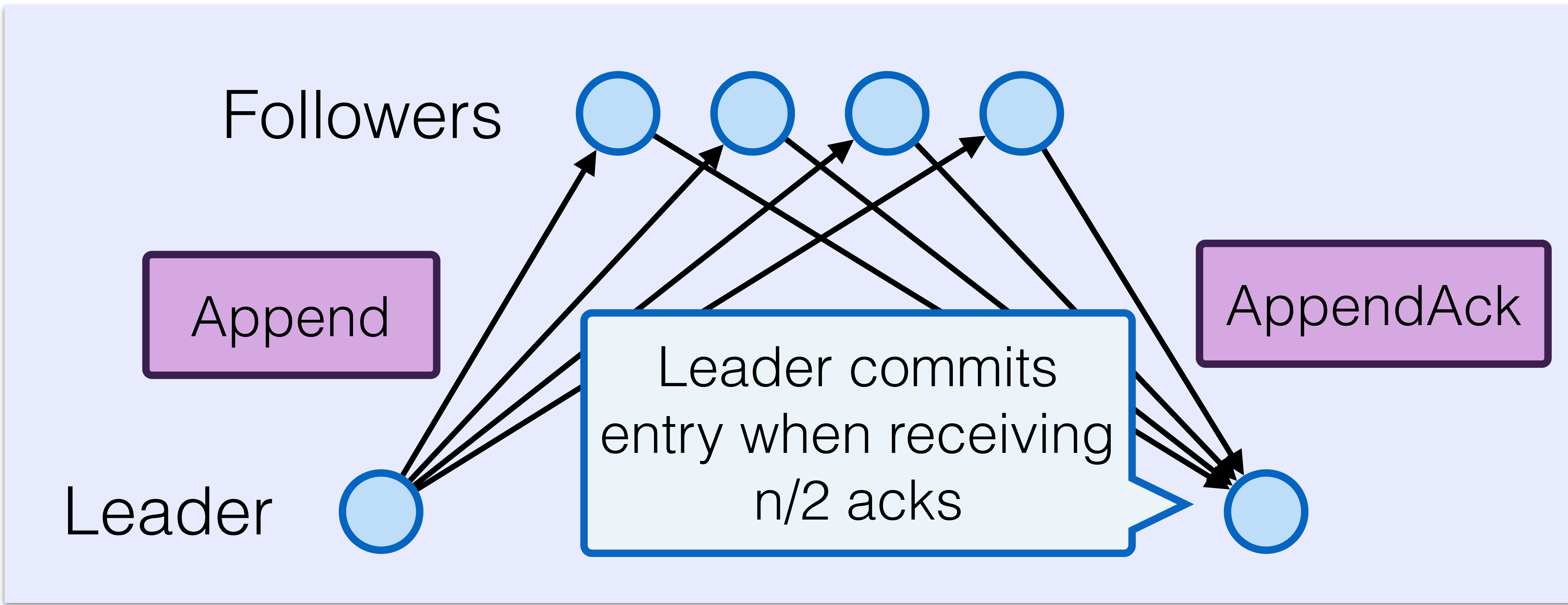
Implementing Raft: Leader Election

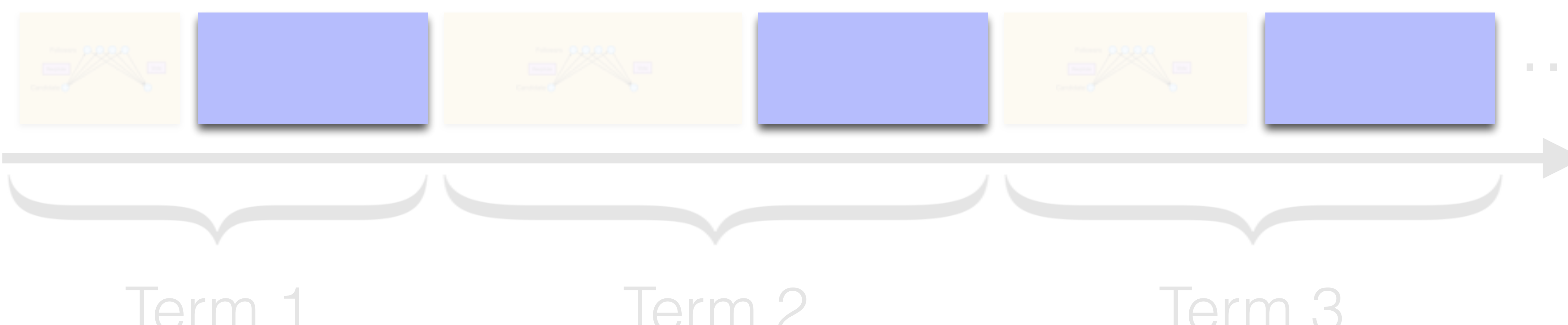


Implementing Raft



Implementing Raft: Log Replication

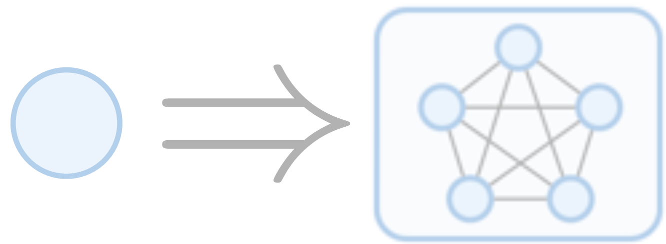




Outline

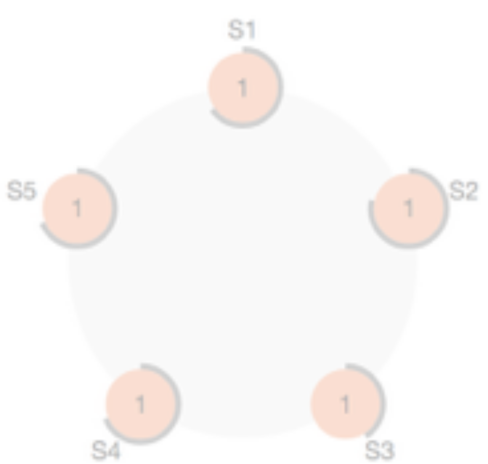
Verification Challenge

state machine replication



Raft Algorithm

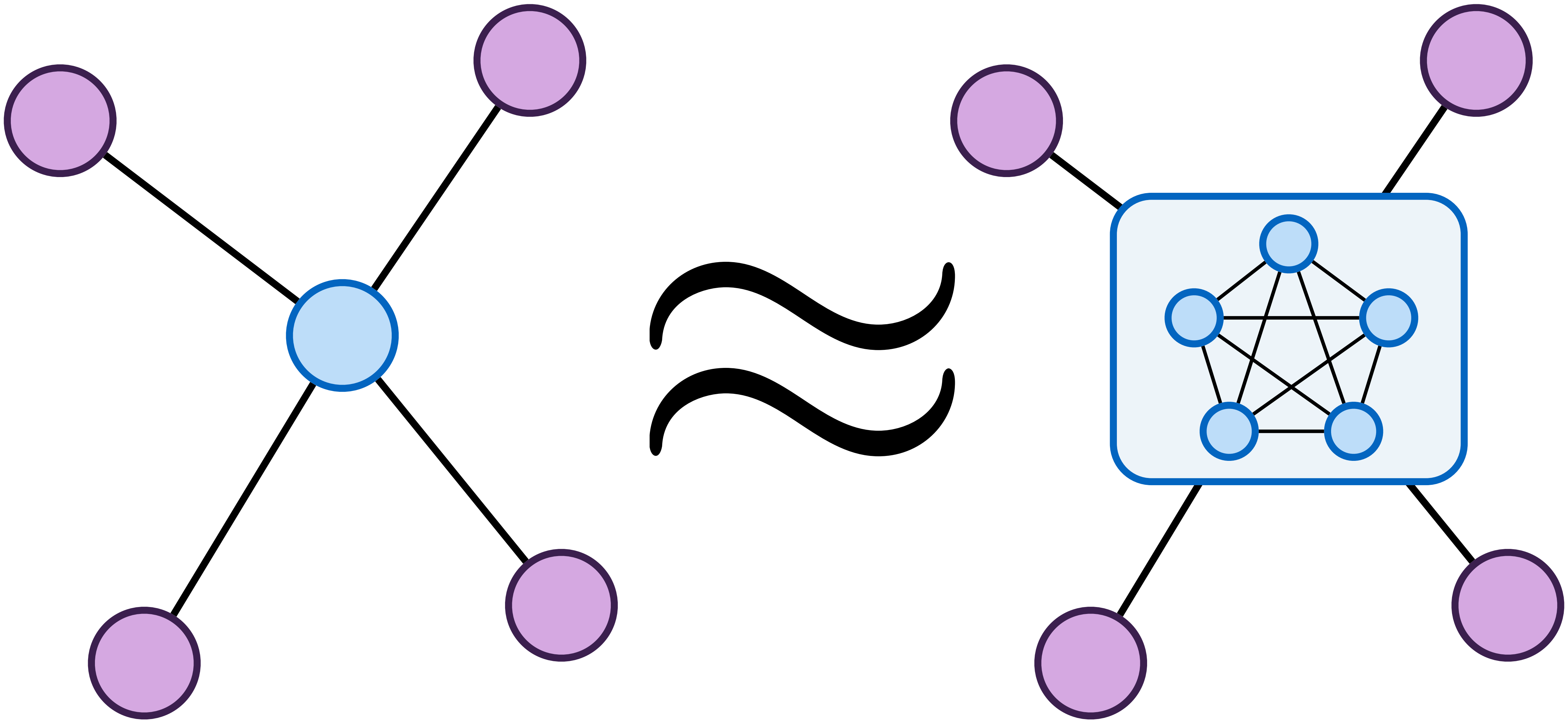
implemented in Verdi



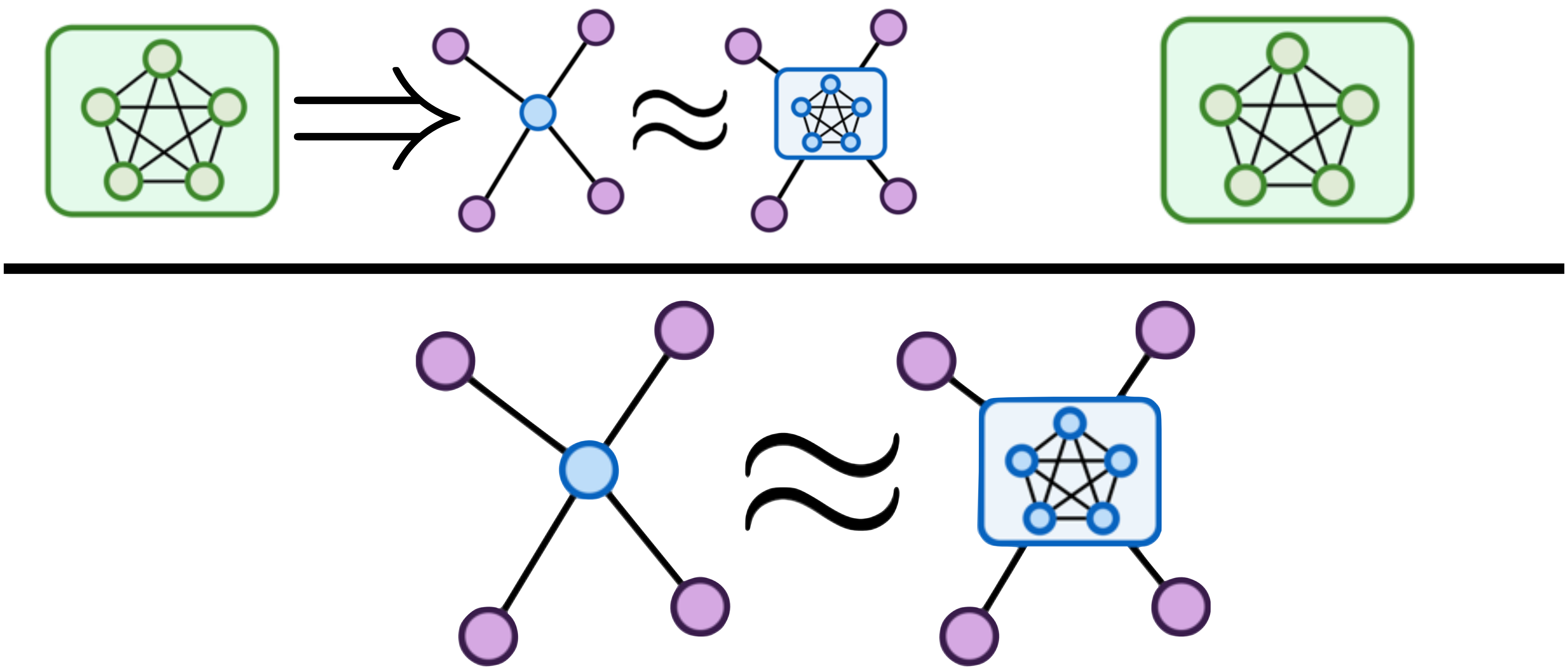
Proof Overview

and lessons learned

Verifying Raft: Show linearizability

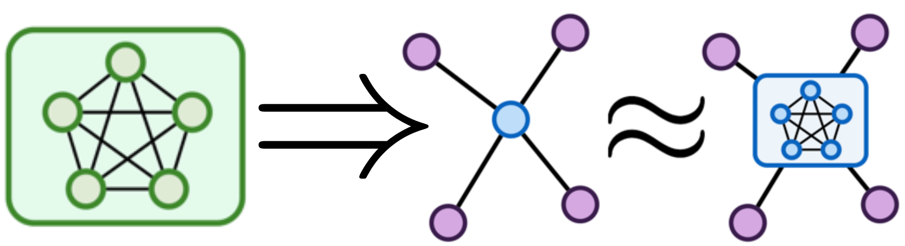


Verifying Raft: Approach

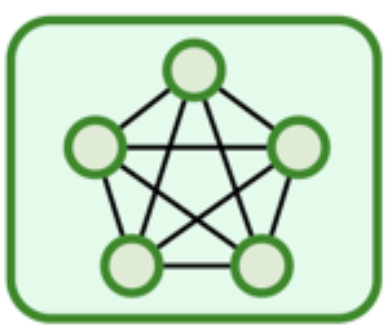


State Machine Safety

Nodes agree about committed entries

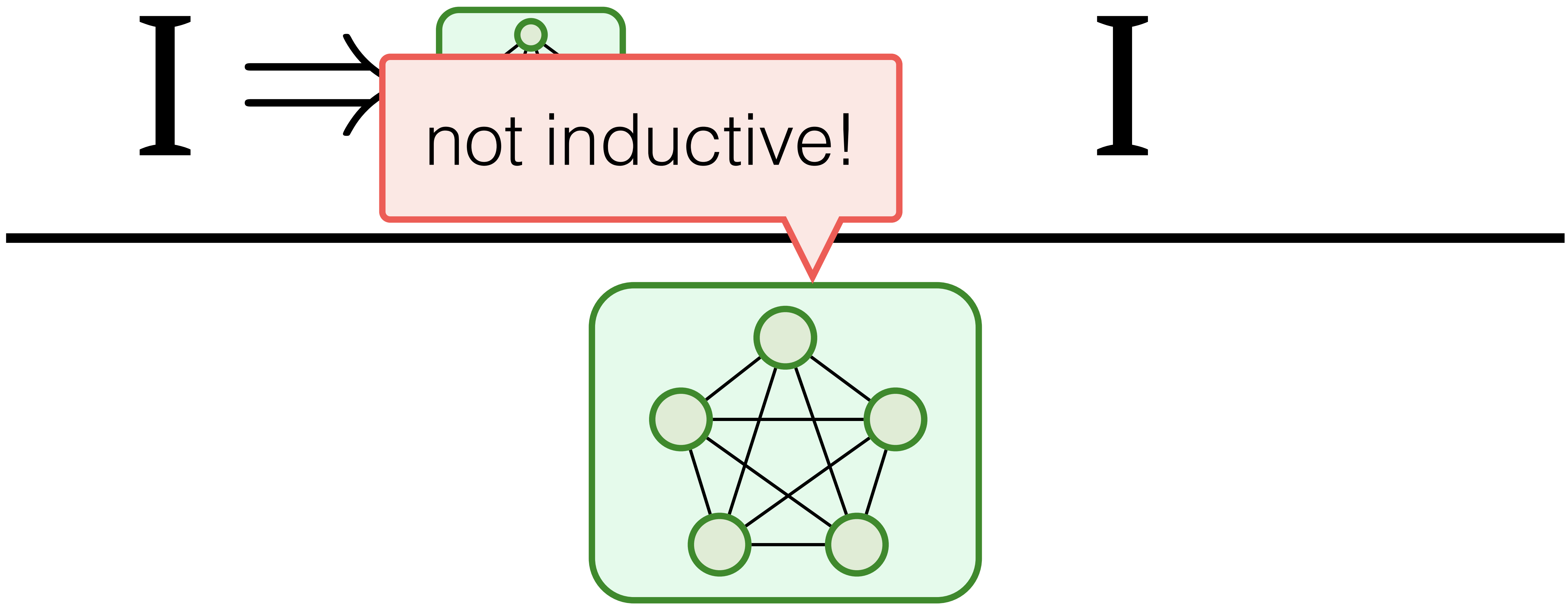


since only committed entries executed

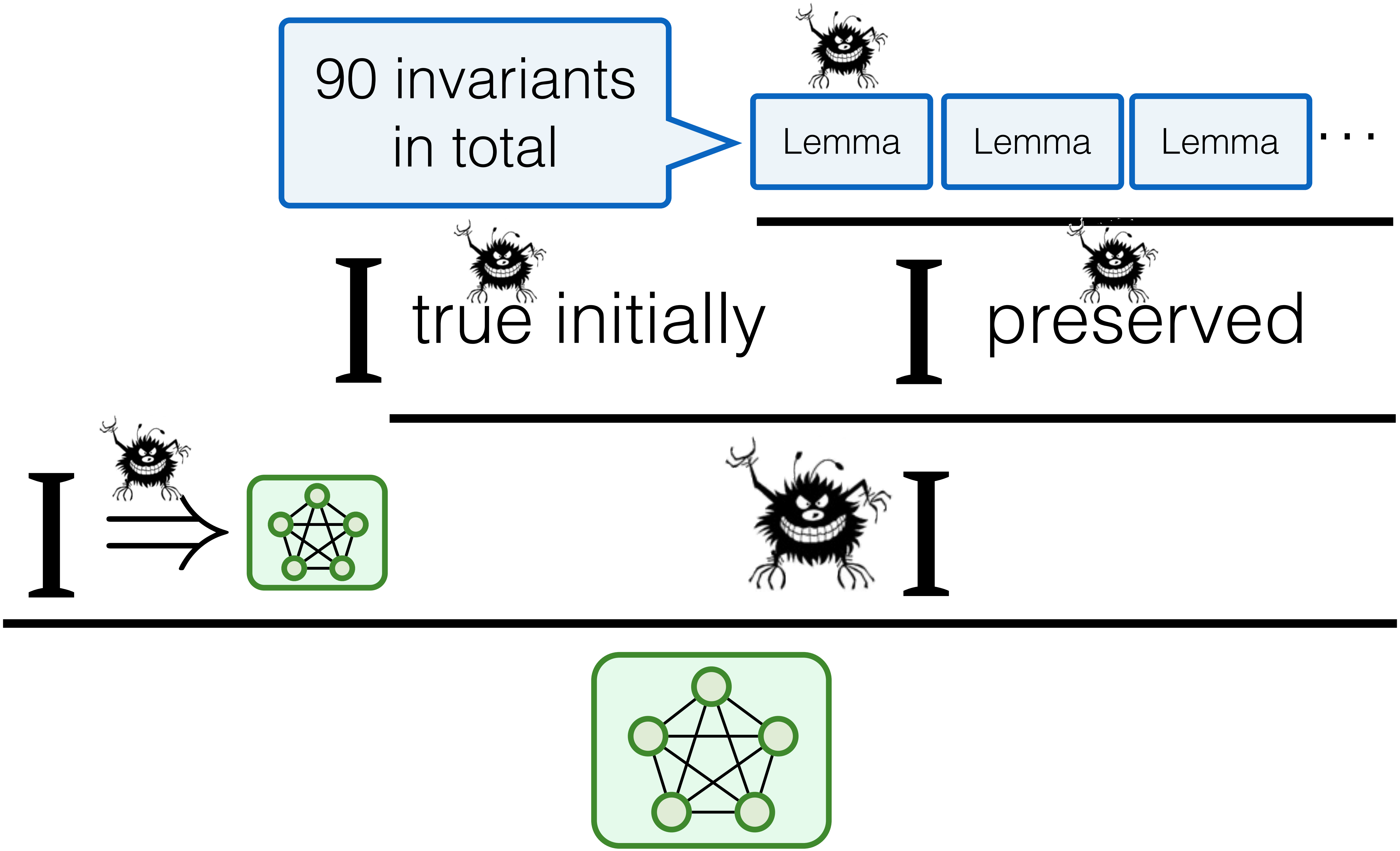


proof by induction on an execution

State Machine Safety: Proof



State Machine Safety: Proof



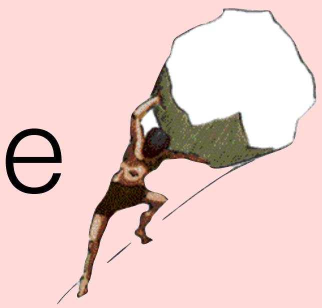
The burden of proof

Re-verification is the primary challenge:

- invariants are not inductive
- not-yet-verified code is wrong
- need additional invariants

The burden of proof

Re-verification is the primary challenge

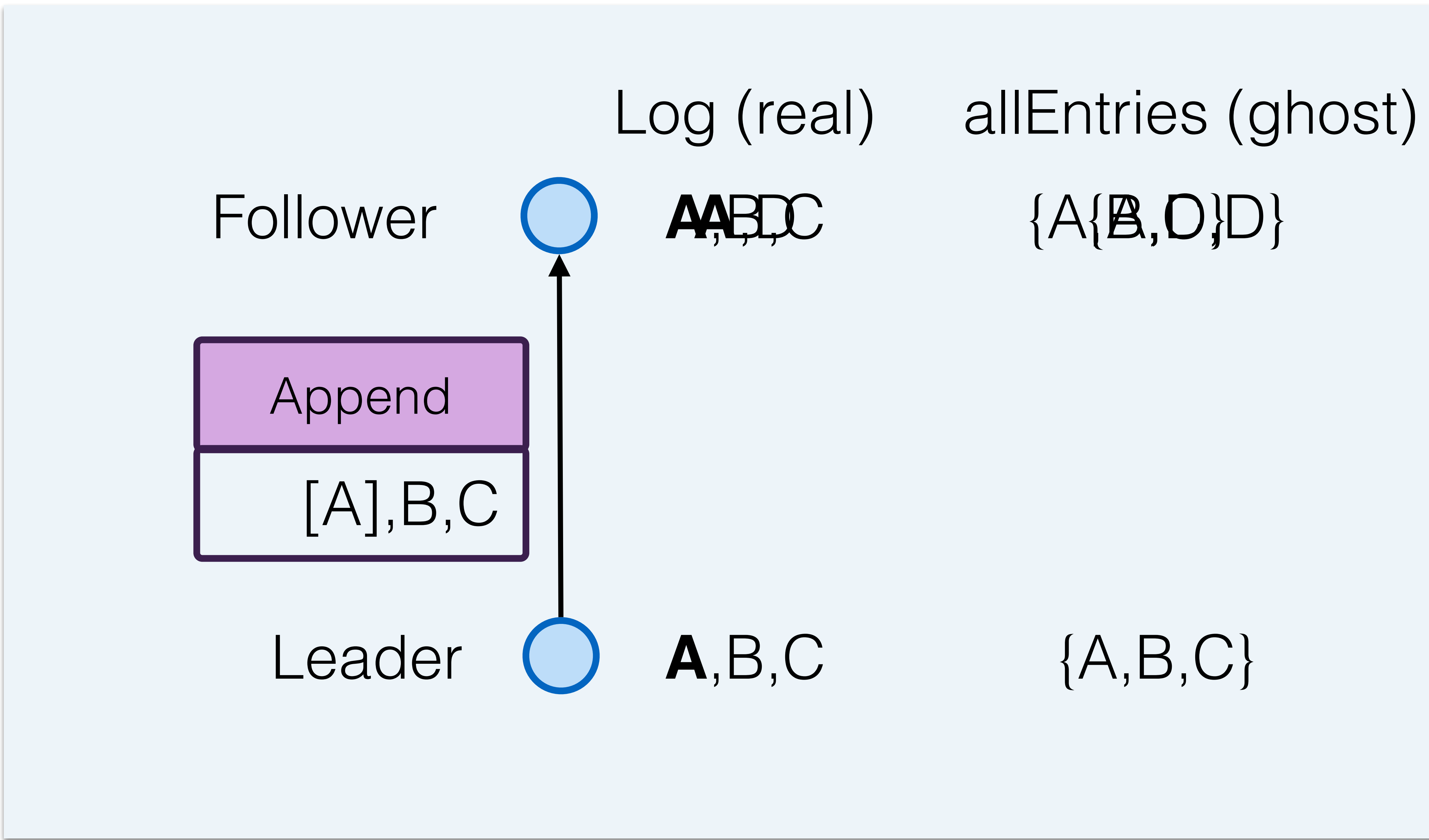


Proof engineering techniques help:

- affinity lemmas
- intermediate reachability
- structural tactics
- information hiding

Ghost State: Example

Capture all entries received by a node



Affinity Lemmas: Example

$$\frac{e \in \text{log} \Rightarrow \quad e.\text{term} > 0}{e \in \text{allEntries} \Rightarrow e.\text{term} > 0}$$

every invariant of entries in logs is invariant of entries in allEntries

Affinity Lemmas: Example

$$\frac{e \in \mathrm{log} \Rightarrow \mathrm{P}\ e}{e \in \mathrm{allEntries} \Rightarrow \mathrm{P}\ e}$$

every invariant of entries in logs is invariant of entries in allEntries

Affinity Lemmas

Ex 1: Relate ghost state to real state

transfer properties once and for all

Ex 2: Relate current messages to past

response => past request

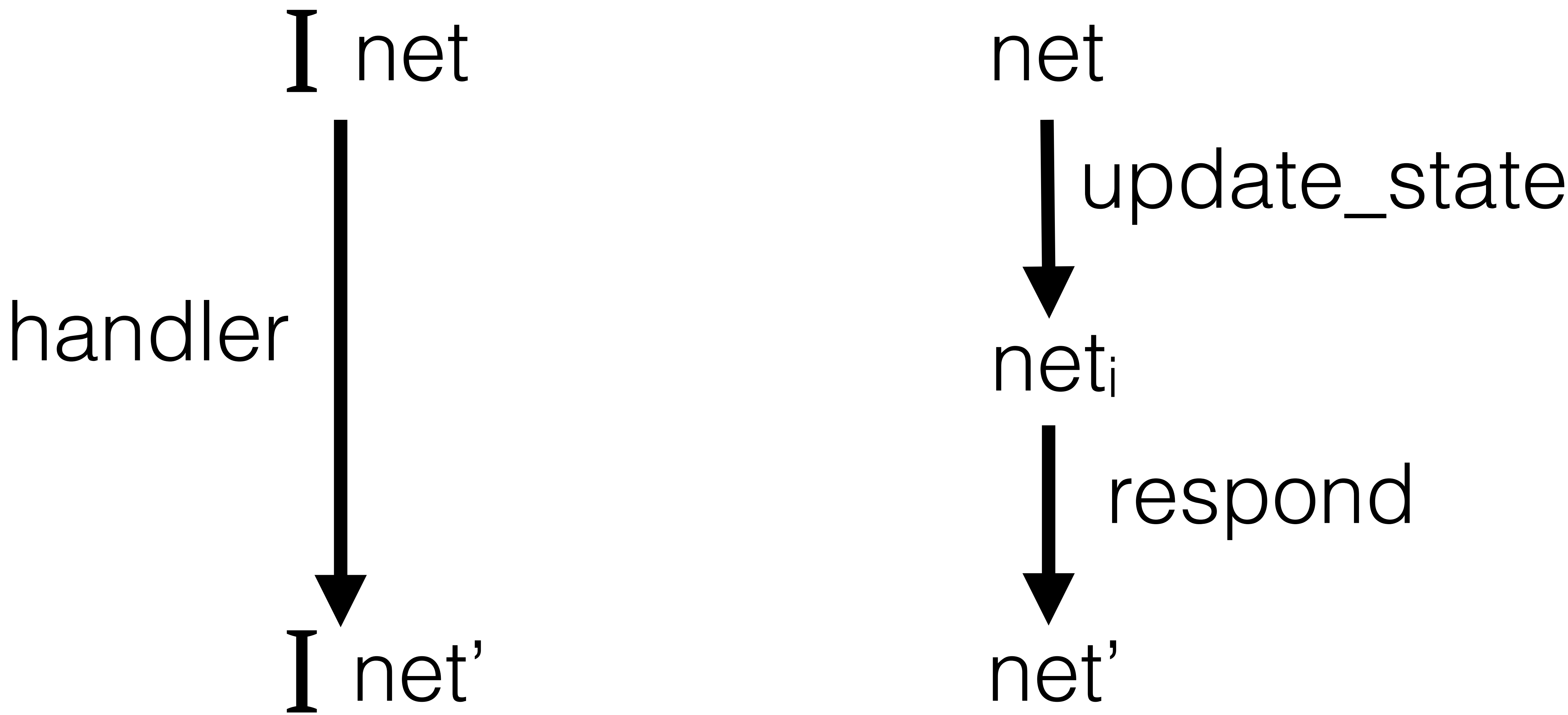
Structured Handlers: Example

handler = update_state ; respond

$$
\begin{tikzcd}
\text{net} \arrow[dd, "\text{handler}"'] & \text{net} \arrow[d, "\text{update\_state}"] \\
 & \text{net}_i \arrow[d, "\text{respond}"] \\
\text{net'} & \text{net'}
\end{tikzcd}
$$

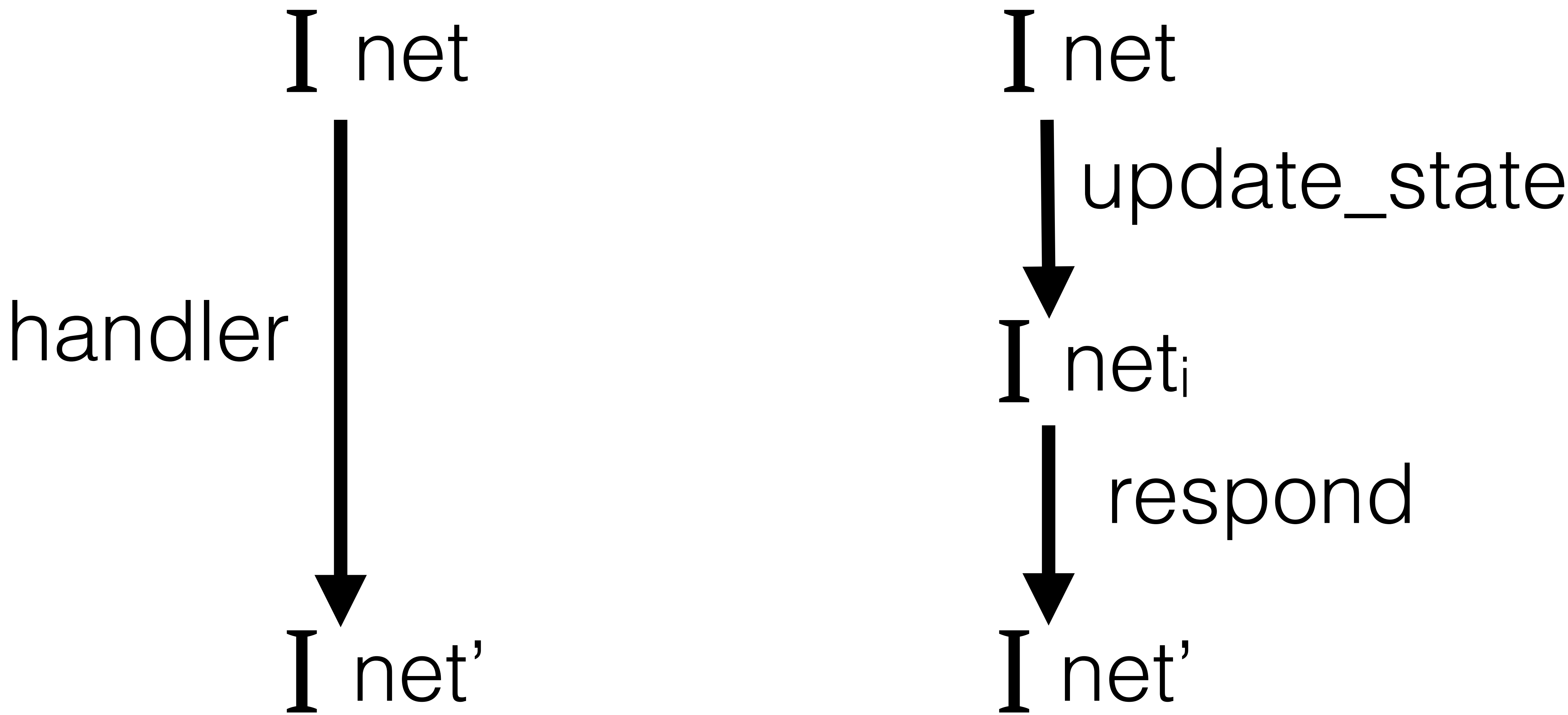
Structured Handlers: Example

handler = update_state ; respond



Structured Handlers: Example

handler = update_state ; respond



The burden of proof

Re-verification is the primary challenge

Proof engineering techniques help:

- affinity lemmas
- intermediate reachability
- structural tactics
- information hiding

Contributions

First formal proof of Raft's safety

first verified implementation!

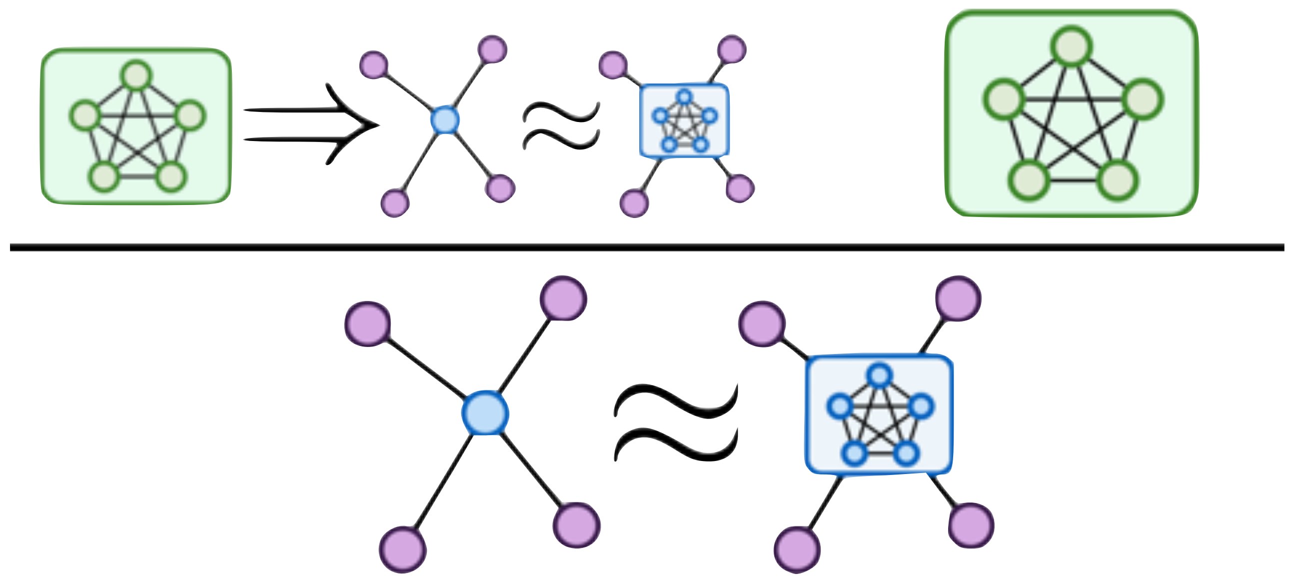
Large-scale Verdi case study

stress test; reverification inevitable

Proof engineering lessons

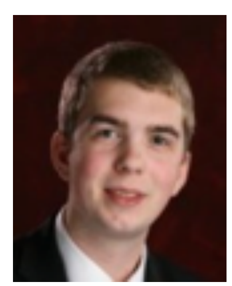
affinity lemmas, etc.

Planning for Change in a Formal Verification of the Raft Consensus Protocol




Doug Woos


James Wilcox


Steve Anton


Zach Tatlock


Michael Ernst


Tom Anderson

University of Washington Computer Science & Engineering

UNIVERSITY of WASHINGTON